STREAMLINING PRODUCT DEPLOYMENT: ENHANCING EFFICIENCY THROUGH KITTING PROCESSES

G. Ayyappan, S. Sankeetha

•

Department of Mathematics, Puducherry Technological University, India.
Department of Mathematics, Saradha Gangadharan College, India.
ayyappanpec@hotmail.com sangeetha.sivarajp@gmail.com

Abstract

Considering a single server with two queues that is prone to unreliability. The server offers a kitting process and performs necessary checks and rectifications when required. The arrival of items follows a Markovian arrival process, while the service is distributed based on a phase type distribution. The incoming products may exhibit issues such as poor quality or defects. If either of the queues is empty, the server is unable to provide the requested service and remains inactive. Furthermore, if all queues are empty, the server goes into a vacation mode. Breakdowns, repairs, instances of customers leaving without service (reneging), and vacation periods are all modeled using an exponential distribution. To gain insights into the performance of the queueing model, various performance metrics are analyzed and represented through 2D and 3D graphs.

Keywords: Markovian Arrival Process, PH distribution, Vacation, Optional service, Breakdown and Repair.

1. Introduction

The Markov arrival process (MAP) is a widely employed modeling approach that captures the dynamic Markov structure underlying point processes. It offers adaptability and versatility, making it suitable for probabilistic models that employ matrix analysis techniques. Neuts [15] made significant contributions by proposing and extensively investigating the flexible nature of Markov point processes. MAP shares similarities with other point processes, including Markov-modulated Poisson processes, phase-like updating processes, and semi-Markov point processes. It enables the simulation of both updating and non-updating models, making it a valuable tool for studying arrival patterns. Chakravarthy [7] has provided in-depth insights and extensive discussions on MAP, specifically focusing on its m-dimensional parameter matrix (D_0, D_1, D_2), where D_0 governs transitions associated with no arrivals and D_1 and D_2 controls alternations related to arrival events. This parameterization allows for effective control and analysis of arrival dynamics in various systems.

Wang et al. [28] presented a framework for optimizing the kitting process in manufacturing. It addresses the challenges of efficiently organizing and sequencing materials required for assembly operations. The authors propose a mathematical model to minimize the overall kitting time, reduce material handling, and improve productivity in manufacturing settings. Yadav et al. [26] focused on optimizing the kitting process in an automotive assembly line. It investigates the challenges associated with kitting and proposes a mathematical model for optimizing the allocation of parts to kits. A hybrid optimization approach is applied that combines genetic algorithms and simulated annealing to minimize the total distance traveled by workers during the kitting process. The study provides insights into improving the efficiency of the kitting process in automotive manufacturing. Ayyappan and Nithya [6] studied a retrial feature that allows customers who experience service unavailability to reattempt service after a certain period. The model considers priority services, where one type of customer is given priority over the other in terms of service. Breakdowns and repairs are differentiated, meaning that the server may require different amounts of time to recover from different types of failures. Synchronized reneging is taken into account, which means that customers may abandon the queue simultaneously if their waiting time exceeds a specific threshold. Additionally, the model incorporates an optional vacation, allowing the server to take breaks during certain periods.

Zhang and Fang [29] introduce a novel optimization algorithm designed to enhance the efficiency of bulk service systems. These systems are frequently encountered in various industries, including manufacturing and transportation, where multiple units of work or customers are processed simultaneously. The primary objective of the proposed algorithm is to minimize service time and decrease waiting times for customers within bulk service systems. To achieve this, the algorithm combines two powerful optimization techniques: stochastic optimization and reinforcement learning. The algorithm works in iterations, continuously refining its policies based on feedback from the system. It collects data on customer arrival patterns, service times, and queue lengths, which are then used to update the stochastic optimization models and reinforce the learned policies. This iterative process allows the algorithm to adapt to dynamic changes in the system and continuously optimize its performance. Li and Li [13] focused on optimizing bulk service systems that involve parallel servers. It addresses the challenges associated with efficiently allocating and coordinating multiple servers to improve system performance. The authors propose novel optimization algorithms and strategies to minimize service time and reduce waiting times for customers.

Smith and Johnson [22] investigated the influence of bulk service providers on the overall performance of supply chains. Also examines how the involvement of bulk service providers affects various aspects of supply chain operations, including efficiency, cost, and customer satisfaction. The impact of bulk service providers on key performance indicators are analyzed such as order fulfillment, inventory management, and lead times. Additionally, it highlights the importance of establishing effective collaboration and coordination mechanisms between bulk service providers and other supply chain stake holders. Also emphasize on the significance of information sharing, communication, and performance monitoring to ensure optimal supply chain performance. Wang et al. [27] presents a hybrid optimization approach specifically tailored for bulk service systems in e-commerce warehouses. The authors combine mathematical modeling, simulation, and metaheuristic algorithms to enhance the efficiency of warehouse operations, such as order picking, packing, and shipping. The proposed approach aims to reduce order fulfillment time and improve customer satisfaction in e-commerce fulfillment centers.

Arun et al.[2] analyzed a bulk service queue with server breakdowns, balking, and reneging. It provides a detailed analysis of the system's performance measures, such as the expected waiting time and the expected queue length, under different scenarios. Sun and Zhang [23] focused on the development of a bulk service system specifically designed for autonomous mobile robots, the growing demand for efficient and flexible service systems in industries where autonomous mobile robots are utilized. These systems involve the simultaneous processing of multiple tasks or requests, and efficient management is crucial to optimize performance and resource utilization. A comprehensive design framework for a bulk service system is proposed that integrates autonomous mobile robots. They outline the key components of the system, including task allocation, robot navigation, and coordination mechanisms. The findings of the study demonstrate the advantages of incorporating autonomous mobile robots into bulk service systems. The proposed design framework provides a blueprint for developing efficient and scalable systems that can adapt to changing demands and optimize resource allocation.

Arivudainambi and Arivudainambi [1] studied a mathematical model for analyzing a bulk service queue with multiple vacations, server breakdowns, and general service times. It provides a detailed analysis of the system's performance measures, such as the expected waiting time and the expected queue length. Li and Zhang [14] proposed an optimal control policies for a bulk service queue with impatient customers and time-varying arrival rates. The proposed policies are designed to minimize the total expected cost, including waiting costs and service costs, under different operating conditions. Saroja and Saravanarajan [20] studied bulk service queueing models with server vacations and feedback controls. It provides a detailed analysis of the system's performance measures, such as the expected waiting time and the expected queue length, under different scenarios. Ayyappan and Meena [5] examined the service rate that gradually declines until degradation is fixed. After completing a certain number of services (K), the degradation is addressed. During the service period, the server may experience a breakdown at any moment, triggering an immediate repair process. Once the service is complete, the server transitions to the close-down process. If there are no customers in the system when the server returns from vacation, the server will wait until a customer arrives. If a customer arrives without a starting failure, the server provides service. However, if there is a starting failure, the server immediately goes into the repair process.

Thottan and DeVeciana [24] presented a vacation model that incorporates autonomous server vacations and customer impatience. The research focuses on analyzing the performance of queueing systems under such conditions and investigates the impact of autonomous server vacations and customer impatience on system efficiency. Huang and Li [8] investigated on optimization of vacation queues that involve multiple vacation periods and general service times. The authors investigate the problem of determining optimal control policies for allocating vacation time and managing service rates in order to optimize various performance measures. They consider system characteristics such as queue length, waiting time, and system utilization. By analyzing the impact of different control policies on the system's performance, the authors provide insights into the efficient management of vacation queues. Their research contributes to the development of strategies for optimizing service allocation and improving the overall efficiency of queueing systems with multiple vacation periods and general service times. Anis et al. [4] explored the analysis of a finite-buffer queue that incorporates server vacations and customer impatience. It investigates the performance measures of the queueing system, including queue length, waiting time, and server utilization. The study provides understanding the enhancement of buffer size, vacation policies, and customer impatience management.

Srinivasan and Sriram [21] analyzed on studying vacation queues where the server is subject to breakdowns and repair. The authors analyze the impact of server breakdowns on the performance of the queueing system. They investigate various performance measures such as queue lengths, waiting times, and server utilization during both normal operation and breakdown periods. The study provides insights into the optimization of repair policies to minimize system downtime and improve overall system performance. By considering the combined effect of vacations and server breakdowns, the authors contribute to the understanding of real-world queueing systems where service interruptions due to breakdowns are common. Kim et al.[10] researched on vacation models that consider customer abandonments. It investigates the impact of customer abandonment behavior on queueing systems during vacation periods. The study provides perception on optimization of vacation policies and customer abandonment management.

Rakesh Kumar et al. [19] examined a single-server Markovian queuing model that incorporated customer impatience, including balking and reneging, alongside a threshold mechanism and customer retention. They employed probability generating functions to analyze the model's transient behavior. Kalyanaraman and Janani [9] addresses a finite population Poisson queue employing a fixed batch service rule. Following each service, the server goes on vacation, regardless of queue size, providing service at a reduced rate during this period. The research calculates system size probabilities, derives performance metrics, and also explores an infinite population model with limited waiting room capacity as a secondary model. Krishnamurthy et al. [11] centers on the examination of a queuing system characterized by its multi-stage bulk service approach and the availability of service in batches. Within this system, incoming customers are initially grouped into batches before undergoing bulk servicing. The research extensively presents mathematical derivations pertaining to performance metrics, including system size, mean waiting time, and mean service time.

Raina Raj and Selvamuthu Dharmaraja [17] introduces an architectural framework that prioritizes energy efficiency within the SAT network, with a particular focus on HAPs. Furthermore, a stochastic model is proposed to account for three distinct states of energy conservation for HAPs, including modes of power conservation, standby, and rest, where energy consumption is minimal or negligible. Upon the arrival of a data packet, HAPs promptly transition to active service mode, ensuring the entire system operates in an active state. Anilkumar and Jose [3] examines a discrete-time inventory model (s, S) is investigated, featuring Bernoulli process customer arrivals and geometrically distributed service and replenishment times. When inventory drops to zero due to customer service or lack of replenishment, the system can accommodate a maximum of k customers, with any excess customers considered lost until replenishment occurs. Rakesh Kumar et al. [18] conducted a comprehensive study examining the utilization of queuing theory in the analysis of cloud computing systems. Their research specifically delved into the phenomenon of task reneging, where requests are dropped from the queue due to user impatience, deadlines, security protocols, or active queue management strategies.

2. Motivation

In a software development company, a team is working on creating a new application that consists of multiple modules and features. Rather than developing and delivering each module individually, they adopt a kitting process to streamline the deployment process and improve efficiency. In this kitting process, each module or feature is treated as a separate item and is placed in a dedicated queue. The server, which represents the deployment team, retrieves the modules from the queues and starts assembling the software kit. They integrate the modules, perform necessary configurations, and ensure compatibility between different components.

Once the kit is assembled, the server performs thorough testing and quality assurance checks to verify the functionality and stability of the software. If any issues are identified, such as bugs or compatibility conflicts, the server rectifies them before proceeding. Once the kit passes the testing phase, it is packaged for release to the end-users or clients. The server ensures that all required documentation, user guides, and support materials are included in the kit before delivering it. By employing the kitting process in software development, the company streamlines the deployment process, reduces errors, and ensures that the end-users receive a comprehensive and well-tested software package.

3. Mathematical Formulation

This model considers two types of arrivals within a system. The first type follows a Markovian arrival process and has infinite capacity, while the second type has a finite capacity of K. The server is responsible for the packing service, which follows a phase type distribution denoted as (α_1, T_1). The equation $T_1^0 + T_1 e = 0$ holds true, where T_1^0 represents a column vector. Once the packing is completed using the kitting process, the server proceeds to verify the checklist for the packed product.

If the checklist is satisfied, the product is deemed ready for the outlet. Otherwise, the server initiates the rechecking and rectification process. This rechecking process follows a phase type distribution denoted as (α_2, T_2). The equation $T_2^0 + T_2 e = 0$ holds true, where T_2^0 represents a column vector. If either of the queues becomes empty, the server remains idle. However, when both queues are empty, the server goes on vacation, with the vacation parameter η following an exponential distribution.

Additionally, the server is subject to breakdown during both the packing service and rechecking, with a breakdown parameter ξ following an exponential distribution. When the server experiences a breakdown while serving, it completes the ongoing service and then enters a repair process with a parameter γ following an exponential distribution. Moreover, the products in both queues are susceptible to reneging, indicated by parameters δ_1 and δ_2, respectively, following an exponential distribution. Reneging can occur due to factors such as lack of quality or defects.

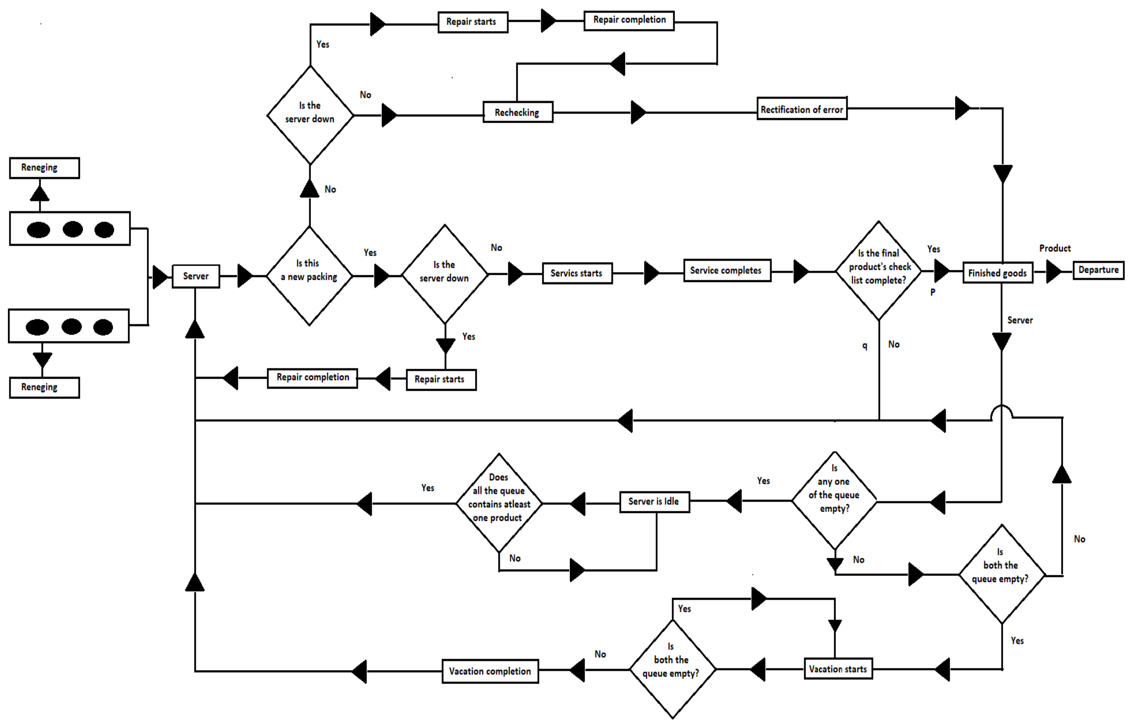

Figure 1: *Schematic Representation of Our Model*

In pursuit of a matrix-geometric solution, the model is explored within the framework of a QBD (Quasi-Birth-Death) process. For a comprehensive exploration of Matrix Analytic Methods, refer to the works of Neuts [16] and Latouche and Ramaswami [12]. The QBD model's state space is formally defined, and an examination of the infinitesimal generator's structure is carried out, leveraging the subsequent notational conventions.

Let

- I_j is the identity matrix of dimension j.
- e_1 is the column vector of dimension $m_1 m_2[(n_1 + n_2)(1 + K) + (4 + 3K)$ with its entries 1.
- $N_1(t)$ indicates the total number of items in the type I queue.
- $N_2(t)$ indicates the total number of items in the type II queue.
- $S(t)$ indicates the position of the server.

where

$$S(t) = \begin{cases} 0, & \text{if server is idle} \\ 1, & \text{if the server is engaged with packing} \\ 2, & \text{if the server is engaged with rework} \\ 3, & \text{if the server faces breakdown while packing} \\ 4, & \text{if the server faces breakdown while rework} \\ 5, & \text{if server is on vacation} \end{cases}$$

- $J_1(t)$ indicates the service phase when the server is engaged with packing.
- $J_2(t)$ indicates it the service phase when the server is engaged with packing.
- $M_1(t)$ indicates the phase of the Markovian Arrival Process for type I queue.
- $M_2(t)$ indicates the phase of the Markovian Arrival Process for type II queue.

Let $\{(N_1(t), N_2(t), S(t), J_1(t), J_2(t), M_1(t), M_2(t)); t \geq 0\}$ represent the continuous time Markov chain for the QBD process with the state space.

$$\Omega = l(0) \cup l(i)$$

where,

$$l(0) = \{(0, j, 0, s_1, s_2) : 0 \leq j \leq K, 1 \leq s_1 \leq m_1, 1 \leq s_2 \leq m_2\}$$

For $i \geq 0$,

$$\begin{aligned} l(i) = \cup & \{(0, j, 1, r_1, s_1, s_2) : 1 \leq j \leq K, 1 \leq r_1 \leq n_1, 1 \leq s_1 \leq m_1, 1 \leq s_2 \leq m_2\} \\ & \cup \{(0, j, 2, r_2, s_1, s_2) : 1 \leq j \leq K, 1 \leq r_2 \leq n_2, 1 \leq s_1 \leq m_1, 1 \leq s_2 \leq m_2\} \\ & \cup \{(0, j, l, s_1, s_2) : 0 \leq j \leq K, 3 \leq l \leq 5, 1 \leq s_1 \leq m_1, 1 \leq s_2 \leq m_2\} \end{aligned}$$

$$For \quad i \geq 1, \quad l(i) = \{(i, 0, 0, s_1, s_2) : 1 \leq s_1 \leq m_1, 1 \leq s_2 \leq m_2\}$$

The infinitesimal matrix generation of the QBD process is given by

$$Q = \begin{bmatrix} B_{00} & B_{01} & 0 & 0 & 0 & 0 & \cdots \\ B_{10} & A_1 & A_0 & 0 & 0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & 0 & 0 & \cdots \\ 0 & 0 & A_2 & A_1 & A_0 & 0 & \cdots \\ \cdots & \cdots & \cdots & \ddots & \ddots & \ddots & \cdots \end{bmatrix}$$

where each of its block matrix are as follows,

$$B_{00} = \begin{bmatrix} b_{00}^{11} & b_{00}^{12} & 0 & 0 & \cdots & 0 & 0 \\ b_{00}^{21} & b_{00}^{22} & b_{00}^{23} & 0 & \cdots & 0 & 0 \\ 0 & b_{00}^{32} & b_{00}^{22} & b_{00}^{23} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & b_{00}^{32} & b_{00}^{M+1M+1} \end{bmatrix}$$

$$b_{00}^{11} = \begin{bmatrix} b_{11}^{011} & \alpha_1 q T_1^0 \otimes I_{m_1 m_2} & \xi I_{m_1 m_2} & 0 & b_{14}^{011} \\ 0 & b_{22}^{011} & 0 & \xi I_{m_1 m_2} & b_{24}^{011} \\ \gamma I_{m_1 m_2} & 0 & b_{33}^{011} & 0 & 0 \\ 0 & \gamma I_{m_1 m_2} & 0 & b_{33}^{011} & 0 \\ 0 & 0 & 0 & 0 & I_{m_2} \otimes D_0 \otimes I_{m_1} \end{bmatrix}$$

$$b_{11}^{011} = I_{m_1} \otimes (T_1 \oplus (D_0 - \xi I_{m_2}))$$
$$b_{14}^{011} = e_{m_2} \otimes \alpha_1 p T_1^0 \otimes I_{m_1}$$
$$b_{22}^{011} = I_{m_2} \otimes (T_2 \oplus (D_0 - \xi I_{m_1}))$$
$$b_{24}^{011} = e_{m_2} \otimes \alpha_2 T_2^0 \otimes I_{m_1}$$
$$b_{33}^{011} = I_{m_2} \otimes (D_0 - \gamma I_{m_1})$$

$$b_{00}^{12} = \begin{bmatrix} 0 & I_{n_1 m_1} \otimes D_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{n_2 m_1} \otimes D_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{m_1} \otimes D_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{m_1} \otimes D_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{m_1} \otimes D_2 \end{bmatrix}$$

$$b_{00}^{21} = \begin{bmatrix} 0 & 0 & 0 & 0 & \delta_2 I_{m_1 m_2} \\ \delta_2 I_{n_1 m_1 m_2} & 0 & 0 & 0 & 0 \\ 0 & \delta_2 I_{n_2 m_1 m_2} & 0 & 0 & 0 \\ 0 & 0 & \delta_2 I_{m_1 m_2} & 0 & 0 \\ 0 & 0 & 0 & \delta_2 I_{m_1 m_2} & 0 \\ 0 & 0 & 0 & 0 & \delta_2 I_{m_1 m_2} \end{bmatrix}$$

$$b_{00}^{22} = \begin{bmatrix} b_{11}^{022} & 0 & 0 & 0 & 0 & 0 \\ \alpha_1 p T_1^0 \otimes I_m & b_{22}^{022} & \alpha_1 q T_1^0 \otimes I_m & \xi I_{mn} & 0 & 0 \\ \alpha_2 T_2^0 \otimes I_m & 0 & b_{33}^{022} & 0 & \xi I_{mn} & 0 \\ 0 & \gamma I_{mn} & 0 & b_{44}^{022} & 0 & 0 \\ 0 & 0 & \gamma I_{mn} & 0 & b_{44}^{022} & 0 \\ 0 & 0 & 0 & 0 & 0 & b_{11}^{022} \end{bmatrix}$$

$$b_{11}^{022} = I_{m_2} \otimes (D_0 - \delta_2 I_{m_1})$$
$$b_{22}^{022} = I_{m_1} \otimes (T_1 \oplus (D_0 - (\xi + \delta_2) I_{m_2}))$$
$$b_{33}^{022} = I_{m_2} \otimes (T_2 \oplus (D_0 - (\xi + \delta_2) I_{m_1}))$$
$$b_{44}^{022} = I_{m_2} \otimes (D_0 - (\gamma + \delta_2) I_{m_1})$$

$$b_{00}^{23} = \begin{bmatrix} I_{m_1} \otimes D_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_{n_1 m_1} \otimes D_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{n_2 m_1} \otimes D_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{m_1} \otimes D_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{m_1} \otimes D_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{m_1} \otimes D_2 \end{bmatrix}$$

$$b_{00}^{32} = \begin{bmatrix} \delta_2 I_{m_1 m_2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \delta_2 I_{n_1 m_1 m_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \delta_2 I_{n_2 m_1 m_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta_2 I_{m_1 m_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta_2 I_{m_1 m_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta_2 I_{m_1 m_2} \end{bmatrix}$$

$$b_{00}^{M+1M+1} = \begin{bmatrix} b_{11}^{0M+1M+1} & 0 & 0 & 0 & 0 & 0 \\ b_{14}^{011} & b_{22}^{0M+1M+1} & b_{23}^{0M+1M+1} & \xi I_{n_1 m_1 m_2} & 0 & 0 \\ b_{24}^{011} & 0 & b_{33}^{0M+1M+1} & 0 & \xi I_{m_1 m_2} & 0 \\ 0 & \gamma I_{m_1 m_2} & 0 & b_{44}^{0M+1M+1} & 0 & 0 \\ 0 & 0 & \gamma I_{m_1 m_2} & 0 & b_{44}^{0M+1M+1} & 0 \\ 0 & 0 & 0 & 0 & 0 & b_{55}^{0M+1M+1} \end{bmatrix}$$

$$b_{11}^{0M+1M+1} = I_{m_2} \otimes (D_0 + D_2 - \delta_2 I_{m_1})$$
$$b_{22}^{0M+1M+1} = I_{m_1} \otimes (T_1 \oplus (D_0 + D_2 - (\xi + \delta_2) I_{m_2}))$$
$$b_{23}^{0M+1M+1} = e_{m_2} \otimes \alpha_1 q T_1^0 \otimes I_{m_1}$$
$$b_{33}^{0M+1M+1} = I_{m_2} \otimes (T_2 \oplus (D_0 + D_2 - (\xi + \delta_2) I_{m_1}))$$
$$b_{44}^{0M+1M+1} = I_{m_2} \otimes (D_0 + D_2 - (\gamma + \delta_2) I_{m_1})$$
$$b_{55}^{0M+1M+1} = I_{m_2} \otimes D_0 + D_2 - \delta_2 I_{m_1}$$

$$B_{01} = \begin{bmatrix} b_{01}^{11} & 0 & 0 & \cdots & 0 \\ 0 & b_{01}^{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & b_{01}^{22} \end{bmatrix}$$

$$b_{01}^{11} = \begin{bmatrix} 0 & I_{n_1 m_2} \otimes D_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{n_2 m_2} \otimes D_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{m_2} \otimes D_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{m_2} \otimes D_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{m_2} \otimes D_1 \end{bmatrix}$$

$$b_{01}^{22} = \begin{bmatrix} I_{m_2} \otimes D_1 & 0 & 0 & 0 & 0 \\ I_{n_1 m_2} \otimes D_1 & 0 & 0 & 0 & 0 \\ 0 & I_{n_2 m_2} \otimes D_1 & 0 & 0 & 0 \\ 0 & 0 & I_{m_2} \otimes D_1 & 0 & 0 \\ 0 & 0 & 0 & I_{m_2} \otimes D_1 & 0 \\ 0 & 0 & 0 & 0 & I_{m_2} \otimes D_1 \end{bmatrix}$$

$$B_{10} = \begin{bmatrix} b_{10}^{11} & 0 & 0 & \cdots & 0 & 0 \\ b_{10}^{21} & b_{10}^{22} & 0 & \cdots & 0 & 0 \\ 0 & b_{10}^{32} & b_{10}^{22} & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & b_{10}^{32} & b_{10}^{22} \end{bmatrix}$$

$$b_{10}^{11} = \begin{bmatrix} 0 & 0 & 0 & 0 & \delta_1 I_{m_1 m_2} \\ \delta_1 I_{n_1 m_1 m_2} & 0 & 0 & 0 & 0 \\ 0 & \delta_1 I_{n_2 m_1 m_2} & 0 & 0 & 0 \\ 0 & 0 & \delta_1 I_{m_1 m_2} & 0 & 0 \\ 0 & 0 & 0 & \delta_1 I_{m_1 m_2} & 0 \\ 0 & 0 & 0 & 0 & \delta_1 I_{m_1 m_2} \end{bmatrix}$$

$$b_{10}^{21} = \begin{bmatrix} e_{m_2} \otimes \alpha_1 p T_1^0 \otimes I_{m_1} & 0 & 0 & 0 & 0 \\ e_{m_2} \otimes \alpha_2 T_2^0 \otimes I_{m_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$b_{10}^{22} = \begin{bmatrix} 0 & \delta_1 I_{n_1 m_1 m_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \delta_1 I_{n_2 m_1 m_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta_1 I_{m_1 m_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta_1 I_{m_1 m_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta_1 I_{m_1 m_2} \end{bmatrix}$$

$$b_{10}^{32} = \begin{bmatrix} 0 & e_{m_2} \otimes \alpha_1 p T_1^0 \otimes I_{m_1} & 0 & 0 & 0 & 0 \\ 0 & e_{m_2} \otimes \alpha_2 T_2^0 \otimes I_{m_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} a_1^{11} & a_1^{12} & 0 & 0 & \cdots & 0 & 0 \\ a_1^{21} & a_1^{22} & a_1^{23} & 0 & \cdots & 0 & 0 \\ 0 & b_1^{32} & a_1^{22} & a_1^{23} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_1^{32} & a_1^{M+1M+1} \end{bmatrix}$$

$$a_1^{11} = \begin{bmatrix} a_{11}^{11} & 0 & 0 & 0 & 0 & 0 \\ b_{14}^{011} & a_{22}^{11} & b_{23}^{0M+1M+1} & \xi I_{m_1 m_2} & 0 & 0 \\ b_{24}^{011} & 0 & a_{33}^{11} & 0 & \xi I_{m_1 m_2} & 0 \\ 0 & \gamma I_{m_1 m_2} & 0 & a_{44}^{11} & 0 & 0 \\ 0 & 0 & \gamma I_{m_1 m_2} & 0 & a_{44}^{11} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66}^{11} \end{bmatrix}$$

$$\begin{aligned} a_{11}^{11} &= I_{m_2} \otimes (D_0 - \delta_1 I_{m_1}) \\ a_{22}^{11} &= I_{m_1} \otimes (T_1 \oplus D_0 - (\xi + \delta_1) I_{m_2}) \\ a_{33}^{11} &= I_{m_2} \otimes (T_2 \oplus D_0 - (\xi + \delta_1) I_{m_1}) \\ a_{44}^{11} &= I_{m_2} \otimes D_0 - (\gamma + \delta_1) I_{m_1} \\ a_{66}^{11} &= I m_2 \otimes D_0 - \delta_1 I_{m_1} \end{aligned}$$

$$a_1^{12} = \begin{bmatrix} I_{m_1} \otimes D_2 & 0 & 0 & 0 & 0 \\ I_{n_1 m_1} \otimes D_2 & 0 & 0 & 0 & 0 \\ 0 & I_{n_2 m_1} \otimes D_2 & 0 & 0 & 0 \\ 0 & 0 & I_{m_1} \otimes D_2 & 0 & 0 \\ 0 & 0 & 0 & I_{m_1} \otimes D_2 & 0 \\ 0 & 0 & 0 & 0 & I_{m_1} \otimes D_2 \end{bmatrix}$$

$$a_1^{21} = \begin{bmatrix} 0 & \delta_2 I_{n_1 m_1 m_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \delta_2 I_{n_2 m_1 m_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta_2 I_{m_1 m_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta_2 I_{m_1 m_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta_2 I_{m_1 m_2} \end{bmatrix}$$

$$a_1^{22} = \begin{bmatrix} a_{11}^{122} & e_{m_2} \otimes \alpha_1 p T_1^0 \otimes I_{m_1} & \xi I_{n_1 m_1 m_2} & 0 & 0 \\ 0 & a_{22}^{122} & 0 & \xi I_{n_1 m_1 m_2} & 0 \\ \gamma I_{m_1 m_2} & 0 & a_{33}^{122} & 0 & 0 \\ 0 & \gamma I_{m_1 m_2} & 0 & a_{44}^{122} & 0 \\ \eta I_{m_1 m_2} & 0 & 0 & 0 & a_{44}^{122} \end{bmatrix}$$

$$\begin{aligned} a_{11}^{122} &= I_{m_1} \otimes (T_1 \oplus D_0 - (\xi + \delta_1 + \delta_2)) I_{m_2} \\ a_{22}^{122} &= I_{m_2} \otimes (T_2 \oplus D_0 - (\xi + \delta_1 + \delta_2)) I_{m_1} \\ a_{33}^{122} &= I_{m_1} \otimes D_0 - (\gamma + \delta_1 + \delta_2) I_{m_2} \\ a_{44}^{122} &= I_{m_1} \otimes D_0 - (\eta + \delta_1 + \delta_2) I_{m_2} \end{aligned}$$

$$a_1^{23} = \begin{bmatrix} I_{n_1m_1} \otimes D_2 & 0 & 0 & 0 & 0 \\ 0 & I_{n_2m_1} \otimes D_2 & 0 & 0 & 0 \\ 0 & 0 & I_{m_1} \otimes D_2 & 0 & 0 \\ 0 & 0 & 0 & I_{m_1} \otimes D_2 & 0 \\ 0 & 0 & 0 & 0 & I_{m_1} \otimes D_2 \end{bmatrix}$$

$$a_1^{M+1M+1} = \begin{bmatrix} a_{11}^{1M+1M+1} & e_{m_2} \otimes \alpha_1 p T_1^0 \otimes I_{m_1} & \xi I_{n_1m_1m_2} & 0 & 0 \\ 0 & a_{22}^{1M+1M+1} & 0 & \xi I_{n_2m_1m_2} & 0 \\ \gamma I_{m_1m_2} & 0 & a_{33}^{1M+1M+1} & 0 & 0 \\ 0 & \gamma I_{m_1m_2} & 0 & a_{33}^{1M+1M+1} & 0 \\ \eta I_{m_1m_2} & 0 & 0 & 0 & a_{44}^{1M+1M+1} \end{bmatrix}$$

$$\begin{aligned} a_{11}^{1M+1M+1} &= I_{m_1} \otimes (T_1 \oplus D_0 + D_2 - (\xi + \delta_1 + \delta_2))I_{m_2} \\ a_{22}^{1M+1M+1} &= I_{m_2} \otimes (T_2 \oplus D_0 + D_2 - (\xi + \delta_1 + \delta_2))I_{m_1} \\ a_{33}^{1M+1M+1} &= I_{m_1} \otimes D_0 + D_2 - (\gamma + \delta_1 + \delta_2)I_{m_2} \\ a_{44}^{1M+1M+1} &= I_{m_1} \otimes D_0 + D_2 - (\eta + \delta_1 + \delta_2)I_{m_2} \end{aligned}$$

$$A_0 = \begin{bmatrix} a_0^{11} & 0 & 0 & \cdots & 0 \\ 0 & a_0^{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & a_0^{22} \end{bmatrix}$$

$$a_0^{11} = \begin{bmatrix} I_{m_2} \otimes D_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_{n_1m_2} \otimes D_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{n_2m_2} \otimes D_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{m_2} \otimes D_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{m_2} \otimes D_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{m_2} \otimes D_1 \end{bmatrix}$$

$$a_0^{22} = \begin{bmatrix} I_{n_1m_2} \otimes D_1 & 0 & 0 & 0 & 0 \\ 0 & I_{n_2m_2} \otimes D_1 & 0 & 0 & 0 \\ 0 & 0 & I_{m_2} \otimes D_1 & 0 & 0 \\ 0 & 0 & 0 & I_{m_2} \otimes D_1 & 0 \\ 0 & 0 & 0 & 0 & I_{m_2} \otimes D_1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} a_2^{11} & 0 & 0 & \cdots & 0 & 0 \\ a_2^{21} & a_2^{22} & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_2^{21} & a_2^{22} \end{bmatrix}$$

$$a_2^{11} = \begin{bmatrix} \delta_1 I_{m_1m_2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \delta_1 I_{n_1m_1m_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \delta_1 I_{n_2m_1m_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta_1 I_{m_1m_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta_1 I_{m_1m_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta_1 I_{m_1m_2} \end{bmatrix}$$

$$a_2^{21} = \begin{bmatrix} e_{m_2} \otimes \alpha_1 p T_1^0 \otimes I_{m_1} & 0 & 0 & 0 & 0 \\ e_{m_2} \otimes \alpha_1 T_2^0 \otimes I_{m_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$a_2^{22} = \begin{bmatrix} \delta_1 I_{n_1 m_1 m_2} & 0 & 0 & 0 & 0 \\ 0 & \delta_1 I_{n_2 m_1 m_2} & 0 & 0 & 0 \\ 0 & 0 & \delta_1 I_{m_1 m_2} & 0 & 0 \\ 0 & 0 & 0 & \delta_1 I_{m_1 m_2} & 0 \\ 0 & 0 & 0 & 0 & \delta_1 I_{m_1 m_2} \end{bmatrix}$$

4. Analysis of the Stability Condition

Determining the stability of a system is crucial to ensure its smooth operation and efficient handling of incoming arrivals. The concept of traffic intensity serves as a key metric in assessing system stability. By comparing the average arrival rate with the average service rate over the long run, we can gauge whether the system is capable of managing the workload effectively. For stability, it is desirable that the traffic intensity remains below 1, indicating that the system can handle the incoming arrivals without becoming overwhelmed.

Analyzing the stability of a Markovian arrival process (MAP) presents unique challenges compared to simpler arrival processes like the Poisson process. This is due to the diverse inter arrival time distributions that MAPs can exhibit. To explore stability conditions in MAPs, researchers employ matrix-analytic methods and simulation-based methods. These approaches involve analyzing matrices and eigenvalues to ascertain the system's stability. Simulation-based methods, in particular, prove valuable when dealing with complex systems that lack analytical solutions, enabling researchers to simulate and study system behavior under varying conditions.

Let A be an irreducible infinitesimal generator matrix of order $m_1 m_2[(n_1+n_2)(1+K)+(4+3K)]$. We can decompose A as $A = A_0 + A_1 + A_2$. The vector $\wp = (\wp_0, \wp_1, \wp_2, ..., \wp_{K+1})$ represents an invariant probability vector. It satisfies the conditions $\wp A = 0$ and $\wp e = 1$, where $\wp e$ denotes the dot product between $\wp$ and the vector e.

$$\wp_0[a_0^{11} + a_1^{11} + a_2^{11}] + \wp_1[a_1^{21} + a_2^{21}] = 0.$$
$$\wp_0[a_1^{21}] + \wp_1[a_0^{22} + a_1^{22} + a_2^{22}] + \wp_2[b_1^{32} + a_2^{21}] = 0.$$
$$\wp_{i-1}[a_1^{23}] + \wp_i[a_0^{22} + a_1^{32} + a_2^{32}] + \wp_{i+1}[b_1^{32} + a_2^{21}] = 0, \text{ for } i = 1 \text{ to } K-1.$$
$$\wp_K[a_1^{23}] + \wp_{K+1}[a_0^{22} + a_1^{K+1,K+1} + a_2^{22}] = 0.$$

Given the normalizing condition $\wp e = 1$, in a stable system, it is necessary that

$$\wp A_0 e_{mn[l(K+1)+1]} < \wp A_2 e_{mn[l(K+1)+1]}.$$
$$\wp_0 a_0^{11} + (\wp_1 + \wp_2 + ... + \wp_{K+1})a_0^{22} < \wp_0 a_2^{11} + (\wp_1 + \wp_2 + ... + \wp_K)a_2^{21} + (\wp_1 + \wp_2 + ... + \wp_{K+1})a_2^{22}.$$

5. The Vector of Invariant Probabilities

The crucial role of capturing the system's steady-state behavior is played by the invariant probability vector, which is symbolically represented as X. In order to obtain the vector X, it is necessary to solve the system of equations represented as $XQ = 0$, while simultaneously ensuring the normalization condition $Xe = 1$. Once the stability requirements are fulfilled, the remaining components of X can be computed using an iterative approach. It is important to emphasize that X can be partitioned into sub-vectors, including X_0 and X_i for $i \geq 1$, which have specific dimensions based on the system's characteristics. The dimension of X_0 is$m_1 m_2[(n_1+n_2)(1+K)+(3+4K)$, while X_i for $i \geq 1$ has a dimension of $m_1 m_2[(n_1+n_2)(1+K)+(4+3K)$. Precisely calculating the values of X_0 and X_i involves considering the unique properties and parameters of the system at hand. The expression for X_i can be represented as:

$$X_i = X_1 R^{i-1}, \quad i = 2, 3, 4, \ldots,$$

Here, R refers to the rate matrix, which serves as the minimal non-negative solution to the matrix quadratic equation.

$$R^2A_2 + RA_1 + A_0 = 0$$

The boundary states, represented as X_0 and X_1, are determined by solving the following equations:

$$X_0B_{00} + X_1B_{10} = 0$$
$$X_0B_{01} + X_1(A_1 + RA_2) = 0$$

These equations are subject to the normalizing condition:

$$X_0e + X_1(I - R)^{-1}e = 1$$

It's worth noting that Latouche and Ramaswamy [12] have improved the computation of the rate matrix R by introducing the Logarithmic Reduction Algorithm. This algorithm simplifies the process of obtaining R, making it more efficient and straightforward.

$$\text{Step 1}: H \leftarrow (-A_1)^{-1}A_0,\ L \leftarrow (-A_1)^{-1}A_2,\ G = L \text{ and } T = H.$$

$$\begin{aligned}\text{Step 2}: U &= HL + LH;\\ M &= H^2;\\ H &= (I - U)^{-1}M;\\ M &= L^2;\\ L &= (I - U)^{-1}M;\\ G &= G + TL;\\ T &= TH;\end{aligned}$$

continue Step 1 until $\|e - Ge\|_\infty < \epsilon$.

$$\text{Step 3}: R = -A_0(A_1 + A_0G)^{-1}.$$

6. Examination of Busy Period

In the context of queueing theory, an essential aspect is the analysis of the busy period. This term refers to the duration that starts when a customer enters an empty queue and concludes when the queue once again becomes vacant. However, when dealing with Quasi-Birth-Death (QBD) processes, a different concept known as the "fundamental period" emerges. The fundamental period characterizes the duration needed for the system to shift from level i to level $i - 1$, where i assumes a value of 2 or greater. It's worth noting that special considerations are needed for boundary states, particularly when i takes on values of 0 or 1. Furthermore, when examining all levels i greater than or equal to 2, it becomes evident that there is a total of $mn[l(1 + K) + 1]$ states. This expression quantifies the number of states associated with each level within the queueing model.

Notations:

- $G_{vv'}(k, x)$ corresponds to the likelihood that the QBD process enters level $u - 1$ at time $t = 0$ after undergoing precisely k leftward transitions and arriving at state (u, v'), under the condition that it initially commenced in state (u, v) at time $t = 0$.
- The transition matrix $\bar{G}_{vv'}(z, s)$ is defined as $\sum_0^\infty z^k \int_0^\infty e^{-sx} dG_{vv'}(k, x)$, where the conditions are $|z| \leq 1$ and $Re(s) \geq 0$. This matrix incorporates a combination of infinite series and integrals to capture the intricate transitions inherent in the QBD process.
- $\bar{G}(z, s)$ takes the form of a matrix $(G_{vv'}(z, s))$ and adheres to the equation $\bar{G}(z, s) = z[sI - A_1]^{-1}A_2 + [sI - A_1]^{-1}A_0\bar{G}^2(z, s)$, representing the interplay among various elements of the QBD process.

- In the context of the first passage time analysis, $G = G_{vv'} = \bar{G}(0,1)$ captures the behavior of the process in the absence of boundary states, providing insights into its performance without considering boundary effects.
- $\bar{G}^{(1,0)}_{(vv')}(k,x)$ is the conditional probability that enters the level 0 from 1 at time $t = 0$.
- $\bar{G}^{(0,0)}_{(vv')}(k,x)$ is the first conditional probability returning to level 0.
- $\Re_{1v}$ denotes the anticipated duration for the first passage between levels u and $u-1$ when the process is in state (u,v) at time $t = 0$.
- $\bar{\Re}_1$ is a column vector composed of the entries $\Re_{1v}$, representing the expected first passage times for different states.
- $\Re_{2v}$ stands for the average number of customers who receive service during the initial passage between levels u and $u-1$ when the process begins in state (u,v) at time $t = 0$.
- $\bar{\Re}_2$ is a column vector composed of the entries $\Re_{2v}$, signifying the average number of service completions during the first passage time for different states.
- $\bar{\Re}_1^{(1,0)}$ represents the average duration for the first passage from level 1 to 0 within the QBD process.
- $\bar{\Re}_2^{(1,0)}$ signifies the average number of completed services during the initial passage from level 1 to 0.
- $\bar{\Re}_1^{(0,0)}$ denotes the average time taken for the first return to level 0 within the QBD process.
- $\bar{\Re}_2^{(0,0)}$ represents the average number of completed services during the initial return to level 0.

The G matrix can be computed using the following expression, utilizing the previously determined rate matrix R obtained through the Logarithmic Reduction Algorithmic technique:

$$G = -[A_1 + RA_2]^{-1}A_2$$

For the boundary states, specifically 1 and 0, we can establish equations satisfied by $\bar{G}^{(1,0)}(z,s)$ and $\bar{G}^{(0,0)}(z,s)$, respectively:

$$\begin{aligned}\bar{G}^{(1,0)}(z,s) &= z\,[sI - A_1]^{-1}\,B_{10} + [sI - A_1]^{-1}\,A_0\bar{G}(z,s)\bar{G}^{(1,0)}(z,s).\\ \bar{G}^{(0,0)}(z,s) &= z\,[sI - B_{00}]^{-1}\,B_{01}\bar{G}^{(1,0)}(z,s).\end{aligned}$$

Since G, $\bar{G}^{(1,0)}(z,s)$, and $\bar{G}^{(0,0)}(z,s)$ are stochastic in nature, we can readily compute moments as follows.

$$\begin{aligned}\Re_1 &= -\frac{\partial}{\partial s}\bar{G}(z,s)|_{s=0,z=1} = -\left[A_0(G+1) + A_1\right]^{-1} e\\ \Re_2 &= \frac{\partial}{\partial z}\bar{G}(z,s)|_{s=0,z=1} = -\left[A_0(G+1) + A_1\right]^{-1} A_2 e\\ \Re_1^{(1,0)} &= -\frac{\partial}{\partial s}\bar{G}^{(1,0)}(z,s)|_{s=0,z=1} = -\left[A_1 + A_0 G\right]^{-1}\left[A_0\Re_1 + e\right]\\ \Re_2^{(1,0)} &= \frac{\partial}{\partial z}\bar{G}^{(1,0)}(z,s)|_{s=0,z=1} = -\left[A_1 + A_0 G\right]^{-1}\left[B_{10}e + A_0\Re_2\right]\\ \Re_1^{(0,0)} &= -\frac{\partial}{\partial s}\bar{G}^{(0,0)}(z,s)|_{s=0,z=1} = -B_{00}^{-1}\left[e + B_{01}\Re_1^{(1,0)}\right]\\ \Re_2^{(0,0)} &= \frac{\partial}{\partial z}\bar{G}^{(0,0)}(z,s)|_{s=0,z=1} = -B_{00}^{-1}B_{01}\Re_2^{(1,0)}.\end{aligned}$$

7. Performance Measures

When a system reaches a steady-state, it signifies that the system has achieved stability and performance measures can be derived and examined. These performance measures play a vital role in evaluating the various aspects of system performance and determining its efficiency and effectiveness. By analyzing these measures, we can gain valuable insights into the system's behavior and identify areas that require improvement to enhance overall performance. Performance measures serve as quantitative indicators that shed light on important system characteristics such as throughput, response time, resource utilization, and reliability. They provide a comprehensive view of how well the system is functioning and can help in assessing its overall effectiveness in meeting desired objectives. By closely monitoring and analyzing performance measures, decision-makers can identify potential bottlenecks, inefficiencies, or areas of improvement within the system. This enables them to make informed decisions and take appropriate actions to optimize system performance, increase productivity, and enhance customer satisfaction.

- Probability the server is idle .
  $P_I = \sum_{j=1}^{K} x_{0j0} + \sum_{i=1}^{\infty} x_{i00}$.
- Probability the server is busy with packing.
  $P_{BP} = \sum_{i=0}^{\infty} \sum_{j=0}^{K} x_{ij1}$.
- Probability the server is busy with rework.
  $P_{BR} = \sum_{i=0}^{\infty} \sum_{j=0}^{K} x_{ij2}$.
- Probability the server is in breakdown while busy with packing.
  $P_{BDP} = \sum_{i=0}^{\infty} \sum_{j=0}^{K} x_{ij3}$.
- Probability the server is in breakdown while busy with rework.
  $P_{BDP} = \sum_{i=0}^{\infty} \sum_{j=0}^{K} x_{ij4}$.
- Probability the server is on vacation.
  $P_V = \sum_{i=0}^{\infty} \sum_{j=0}^{K} x_{ij5}$.
- Expected system size
  $E_{System} = x_1[(I-R)^{-2}]e_1$.

8. Cost Analysis

Let us introduce a cost associated with different system management metrics for our model of interest. We can then formulate a cost function, TC, which takes these metrics into account.
$TC = CH * E_{system} + P_V * CV + P_I * CI + P_{BP} * CBP + P_{BR} * CBR + P_{BDP} * CBDP + P_{BDR} * CBDR + \mu_1 * C1 + \mu_2 * C2 + \gamma * C3$
where

- TC-Total cost of the system per unit time.
- CH-Customer holding cost in the system per unit time.
- CV - Cost when the server is on vacation per unit time.
- CI - Cost when the server is idle per unit time.
- CBP - Cost when the server is busy with packing per unit time.
- CBR - Cost when the server is busy with rework per unit time.
- CBDP- Cost when the server faces breakdown while packing per unit time.

- CBDR- Cost when the server faces breakdown while rework per unit time.
- C1 -Cost afforded for packing service by the server per unit time per unit time.
- C2 - Cost afforded for rework service by the server per unit time per unit time.
- C3 - Cost afforded for carrying out the repair process per unit time.

9. Numerical Analysis

In this section, we will delve into the qualitative behavior of the model through a series of illustrations that include both numerical and graphical representations. By manipulating various model parameters, such as the arrival process and service time distribution, we aim to gain a deeper understanding of how these parameters affect the model's behavior. Input data for these parameters will be drawn from three sets of values available in the literature, allowing us to examine a wide range of scenarios and explore the model's response to different parameter settings. Through these illustrations, we will shed light on the dynamics and trends exhibited by the model as we vary the model parameters, helping us gain insights into its behavior in different scenarios.

Erlang of order 2 (ERL-A)

$$D_0 = \begin{bmatrix} -5 & 5 & 0 & 0 & 0 \\ 0 & -5 & 5 & 0 & 0 \\ 0 & 0 & -5 & 5 & 0 \\ 0 & 0 & 0 & -5 & 5 \\ 0 & 0 & 0 & 0 & -5 \end{bmatrix}; D_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \end{bmatrix}; D_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Exponential (Exp-A)

$$D_0 = \begin{bmatrix} -1 \end{bmatrix}; D_1 = \begin{bmatrix} 0.6 \end{bmatrix}\ D_2 = \begin{bmatrix} 0.4 \end{bmatrix}$$

Hyperexponential (HYP-EXP-A)

$$D_0 = \begin{bmatrix} -1.90 & 0 \\ 0 & -0.19 \end{bmatrix}; D_1 = \begin{bmatrix} 1.026 & 0.114 \\ 0.1026 & 0.0114 \end{bmatrix}\ D_2 = \begin{bmatrix} 0.684 & 0.076 \\ 0.0684 & 0.0076 \end{bmatrix}$$

Given that Varghese et al. [25] has suggested three phase type distributions for the service process, we will consider these distributions in our analysis. These phase type distributions, which have been proposed by Chakravarthy [7] and documented in the literature, will serve as the basis for our examination of the model's behavior. By incorporating these distributions into our analysis, we aim to gain a deeper understanding of how the model performs under different service time distribution settings and how it responds to varying parameters associated with these distributions. This will enable us to assess the qualitative behavior of the model and uncover any patterns or trends that emerge as we explore these three phase type distributions.

Erlang of order 2 (ERL-S)

$$\alpha_1 = \alpha_2 = (1, 0); T_1 = T_2 = \begin{bmatrix} -2 & 2 \\ 0 & -2 \end{bmatrix}$$

Exponential (Exp-A)

$$\alpha_1 = (1); T_1 = \begin{bmatrix} -1 \end{bmatrix}$$

$$\alpha_2 = (1); T_2 = \begin{bmatrix} -1 \end{bmatrix}$$

Hyperexponential (HYP-EXP-A)

$$\alpha_1 = (0.3, 0.7);\ T_1 = \begin{bmatrix} -9 & 3 \\ 2 & -8 \end{bmatrix}$$

$$\alpha_2 = (0.4, 0.6);\ T_2 = \begin{bmatrix} -12 & 6 \\ 5 & -10 \end{bmatrix}$$

Illustration 1:
In this analysis, we examine the implications of the reneging rate (δ_1) of the customers on the expected system size (E_{system}) for various combinations of service and arrival times. We consider specific parameter values, including $\lambda = 2$, $\mu_1 = 5$, $\mu_2 = 6$, $\xi = 1$, $\gamma = 3$, $\delta_2 = 1$, $\eta = 4$, $p = 0.3$, and $q = 0.7$. The observations derived from Table 1 to 3 are outlined below.

- As the reneging rate increases, more customers choose to leave the system without completing their service requests. This results in a lower number of customers in the system at any given time, leading to an decrease in the expected system size.
- When customers renege at a higher rate, the system experiences a shorter average waiting time and lower congestion due to customers leaving before being served. This decrease congestion leads to only few customers remaining in the system, resulting in a lower expected system size.

Illustration 2:
In this analysis, we examine the effects of the vacation rate (ξ) and service rate (μ_1) of the server on the expected system size $(E_s ystem)$. We consider various combinations of service and arrival times and use specific parameter values, including $\lambda = 2$, $\mu_1 = 6$, $\gamma = 3$ $\delta_1 = 1$, $\delta_2 = 1$ $\eta = 4$, $p = 0.3$, and $q = 0.7$. The observations derived from Figure 29-37 are outlined below.

- When both the vacation rate (ξ) and service rate (μ_1) increase, it generally leads to a decrease in the expected system size. This means that, on average, there will be fewer customers present in the system at any given time.
- An increase in the vacation rate (ξ) implies that the availability of the server increases. Similarly, an increase in the service rate (μ_1) means that the server can process customer requests at a faster pace. When both the vacation rate and service rate increase, the server has a reduced overall availability for serving customers due to more frequent breaks.
- These observations highlight the varying impacts of vacation rate and service rate on the projected system size across different arrival and service times. Erlang arrivals show the most significant reduction in system size, followed by exponential arrivals, while hyper exponential arrivals display a slower rate of decrease.

Table 1: *Renege rate* (δ_1) *vs Expected System Size - ERL-A*

	service		
δ_1	Erlang	Exponential	Hyperexponential
1.0	2.689867713	2.764059228	2.776027059
1.1	2.680737416	2.74942792	2.765549157
1.2	2.672576038	2.734797612	2.755075026
1.3	2.665225627	2.720166304	2.744593354
1.4	2.658647922	2.705534995	2.724115453
1.5	2.647557358	2.690903687	2.703637552
1.6	2.640837634	2.676272379	2.683159655
1.7	2.625331925	2.661641071	2.672681749
1.8	2.611886634	2.647009763	2.662203847
1.9	2.591670138	2.632378454	2.651725946

Table 2: *Renege rate (δ_1) vs Expected System Size - EXP-A*

	service		
δ_1	Erlang	Exponential	Hyperexponential
1.0	2.732538773	2.776538896	2.815784456
1.1	2.721217834	2.766061989	2.794520162
1.2	2.710069838	2.755584087	2.777635494
1.3	2.709039454	2.745106186	2.763847123
1.4	2.698116376	2.734628284	2.752344555
1.5	2.689888555	2.724150383	2.742586259
1.6	2.681785353	2.713672482	2.734194049
1.7	2.673681963	2.703194585	2.726894449
1.8	2.665578647	2.699716679	2.720483688
1.9	2.657475344	2.682238777	2.714806803

Table 3: *Renege rate (δ_1) vs Expected System Size - HYP-EXP-A*

	service		
δ_1	Erlang	Exponential	Hyperexponential
1.0	2.857181106	2.902141938	3.011784445
1.1	2.832316046	2.882431771	2.969464924
1.2	2.813250935	2.866914417	2.945390736
1.3	2.798124579	2.854332628	2.926737224
1.4	2.785797988	2.843905244	2.911866499
1.5	2.775536658	2.835114586	2.899730666
1.6	2.766845678	2.827598651	2.889632399
1.7	2.759379139	2.821103707	2.881091624
1.8	2.752887573	2.815428859	2.873767852
1.9	2.747186259	2.797494693	2.867413281

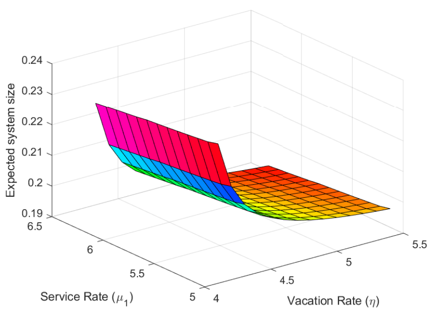

Figure 2: *Vacation rate (η), Service rate (μ_1) vs Expected system size - Ek/Ek/1*

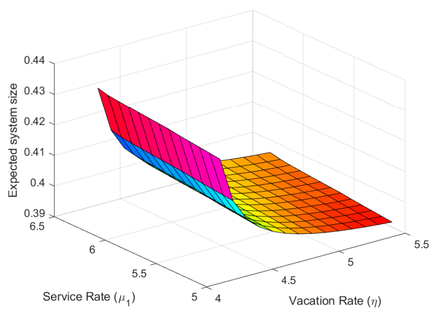

Figure 3: *Vacation rate (η), Service rate (μ_1) vs Expected system size - Ek/M/1*

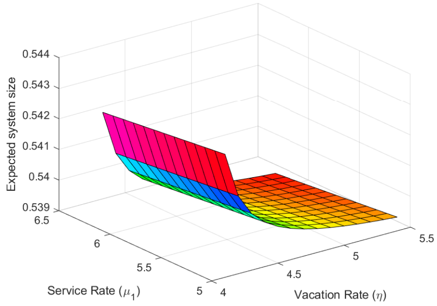

Figure 4: *Vacation rate* (η), *Service rate* (μ_1) *vs Expected system size - Ek/Hk/1*

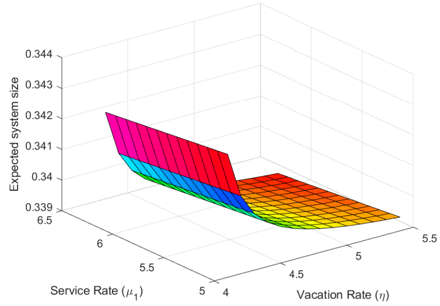

Figure 5: *Vacation rate* (η), *Service rate* (μ_1) *vs Expected system size - M/Ek/1*

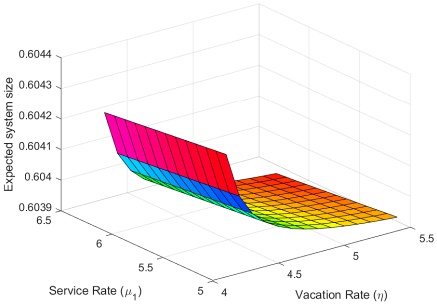

Figure 6: *Vacation rate* (η), *Service rate* (μ_1) *vs Expected system size - M/M/1*

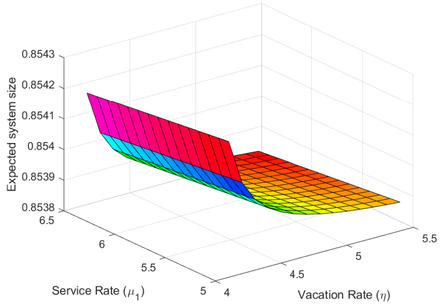

Figure 7: *Vacation rate* (η), *Service rate* (μ_1) *vs Expected system size - M/Hk/1*

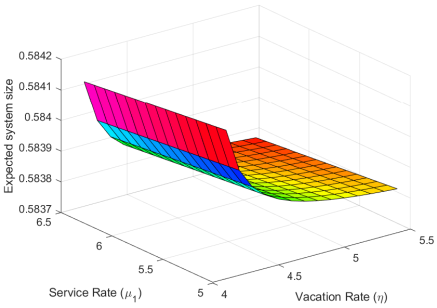

Figure 8: *Vacation rate* (η), *Service rate* (μ_1) *vs Expected system size - Hk/Ek/1*

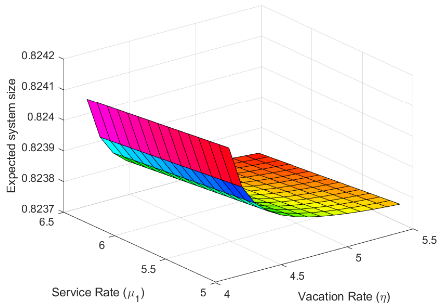

Figure 9: *Vacation rate* (η), *Service rate* (μ_1) *vs Expected system size - Hk/M/1*

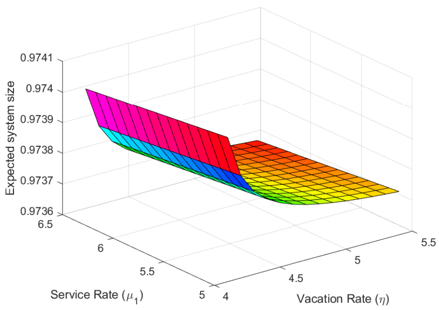

Figure 10: *Vacation rate* (η), *Service rate* (μ_1) *vs Expected system size - Hk/Hk/1*

10. Conclusion

In conclusion, our model encompasses a complex system involving a single server managing two queues, each susceptible to various uncertainties. We have meticulously detailed the arrival and service processes, highlighting the critical phases and distribution patterns that govern them. The model accounts for the inherent unpredictabilities, such as server breakdowns, repairs, customer reneging, and vacation periods. By examining both infinite and finite capacity arrivals, we have provided a comprehensive framework for analyzing the performance and reliability of this intricate system. This model can serve as a valuable tool for optimizing operations, enhancing service quality, and minimizing disruptions in scenarios where such intricate dynamics are at play.

Broadening the system's scope to accommodate intricate service time patterns mirroring real-world complexities holds the potential for a more profound comprehension of service dynamics. Upcoming research endeavors will center on refining scheduling strategies and computational methods for handling batch arrivals, server disruptions, repair processes, bulk services, and the involvement of multiple service providers. These initiatives seek to minimize customer waiting intervals, optimize resource distribution, and elevate overall system effectiveness, with the ultimate goal of enhancing the applicability of such systems across diverse domains.

References

[1] Arivudainambi, K. and Arivudainambi, V. (2018). Performance analysis of a bulk service queue with multiple vacations, server breakdowns, and general service times, *Journal of Industrial Engineering and Management Science*, 1(1):49–60.

[2] Arun, P., Arivudainambi, K., and Arivudainambi, V. (2017). Analysis of a bulk service queue with server breakdowns, balking, and reneging, *Journal of Industrial Engineering International*, 13(4):523–532.

[3] Anilkumar, M. P. and Jose, K. P. (2022). Analysis is a discrete time queueing-inventory model with back-order of items, *3C Empresa. Investigación y pensamiento crítico.*, 11(2):50–62.

[4] Anis, B., Arbia, Y. and Hamdi, M. (2021). Analysis of a finite-buffer queue with server vacations and impatience, *International Journal of Systems Science: Operations and Logistics*, 8(1):111–126.

[5] Ayyappan, G. and Meena, S. (2023). Phase type queueing model of server vacation, repair, and degrading service with breakdown, starting failure and closedown, *Reliability: Theory and Applications*, 18(2):464–483

[6] Ayyappan, G. and Nithya, S. (2023). Analysis of $M^{[X_1]}, M^{[X_1]}, /G_1, G_2/1$ retail queue with priority service, differentiate breakdown, repair, synchronized reneging and optional vacation, *Reliability: Theory and Applications*, 18(2):376–391

[7] Chakravarthy, SR. (2010). Markovian Arrival Processes, *Wiley Encyclopedia of Operations Research and Management Science*, DOI.org/10.1002/9780470400531.eorms0499

[8] Huang, H. and Li, W. (2020). Optimal control of vacation queues with multiple vacations and general service times, *Applied Mathematics and Computation*, DOI: 10.1016/j.amc.2020.125220

[9] Kalyanaraman, R. and Janani, G. (2023). Finite Population and Finite Capacity Single Server Batch Service Queues With Compulsory Working Vacation, *International Journal of Mathematics in Operational Research*, DOI: 10.1504/IJMOR.2023.10057466 .

[10] Kim, D., Kim, J. and Whitt, W. (2021). Vacation models with abandonments, *Management Science*, 67(4):2361–2385.

[11] Krishnamoorthy, A., Joshua, A.N. and Vishnevsky, V. (2021). Analysis of a k-Stage Bulk Service Queuing System with Accessible Batches for Service, *Mathematics* , DOI.org/10.3390/math9050559

[12] Latouche, G. and Ramaswami, V. (1999). Introduction to Matrix Analytic Methods in Stochastic Modeling, *American Statistical Association,Virginia and the Society for Industrial and Applied Mathematics, Pennsylvania*, DOI.org/10.1137/1.9780898719734.

[13] Li, H. and Li, X. (2021). Optimization of Bulk Service Systems with Parallel Servers, *Computers and Industrial Engineering*, DOI.org/10.1016/j.cie.2020.107207

[14] Li, Y., and Zhang, G. (2017). Optimal control of a bulk service queue with impatient customers and time-varying arrival rates, *Journal of Systems Science and Complexity*, 30(5):1165–1180.

[15] Neuts, M. F. (1979). A versatile Markovian point process, *Journal of Applied Probability*, 16(4):764–779

[16] Neuts, M.F. (1981). Matrix-Geometric Solutions in Stochastic Models An Algorithmic Approach, *Johns Hopkins University Press, Baltimore and London*.

[17] Raina Raj and Selvamuthu Dharmaraja (2023). Stochastic Modelling of Multi-Layer Hap-Leo Systems in 6g for Energy Saving: An Analytical Approach, *Computer Communications*, DOI:10.1016/j.comcom.2023.07.037

[18] Rakesh Kumar et al. (2021). Performance Analysis of a Cloud Computing System using Queuing Model with Correlated Task Reneging, *Journal of Physics Conference Series*, 2091(1):1–6

[19] Rakesh Kumar, Sapana Sharma and Sherif I. Ammar (2019). Transient and steady-state analysis of a queuing system having customers' impatience with threshold, *RAIRO - Operations Research*, 53(5):1861–1876

[20] Saroja, V. and Saravanarajan, S. (2018). Bulk service queueing models with server vacations and feedback controls, *Journal of Industrial Engineering and Management Science*, 1(2):89–100.

[21] Srinivasan, A. and Sriram, S. S. (2020). Analysis of vacation queues with server subject to breakdowns and repair, *International Journal of Applied Mathematics and Computer Science*, 30(4):637–652.

[22] Smith, E. and Johnson, J. (2022). An Analysis of the Impact of Bulk Service Providers on Supply Chain Performance, *Journal of Business Logistics*, 43(1):1–16.

[23] Sun, Y. and Zhang, J. (2022). Designing a Bulk Service System for Autonomous Mobile Robots, *IEEE Transactions on Robotics*, 38(1):247–258.

[24] Thottan, M. and DeVeciana, G. (2021). A vacation model with autonomous server vacations and customer impatience, *INFORMS Journal on Computing*, 33(3):982–999.

[25] Varghese, J., Chakravarthy, SR. and Krishnamoorthy, A. (2012). On a customer induced interruption in a service system, *J Stoch Anal Appl*, 30(6):949–962.

[26] Yadav, S. et al. (2020). Optimization of Kitting Process in an Automotive Assembly Line, *Production and Manufacturing Research*, 8(1):551–570.

[27] Wang, Y. et al. (2021). A Hybrid Approach for Optimizing Bulk Service Systems in E-commerce Warehouses, *Computers and Industrial Engineering*, DOI.org/10.1016/j.cie.2021.107272

[28] Wang, L. et al. (2021). A Framework for Kitting Process Optimization in Manufacturing, *Journal of Manufacturing Systems*, 60:487–501.

[29] Zhang, X. and Fang, Y. (2022). Efficient Optimization Techniques for Bulk Service Systems, *IEEE Transactions on Automation Science and Engineering*, 19(1):265–277.