Methodol Comput Appl Probab (2026) 28:10
https://doi.org/10.1007/s11009-026-10244-3

RESEARCH

Analysis of *BMAP/PH/c* Retrial Queue with Threshold-Controlled Retrials and Synchronous Working Vacation

Natarajan Aishwarya[1] · **Agassi Melikov**[2] · **Govindan Ayyappan**[1]

Received: 20 September 2025 / Revised: 2 January 2026 / Accepted: 16 January 2026

Abstract

This work analyzes a multi-server retrial queueing system regulated by a batch Markovian arrival process and phase-type service times, incorporating a control mechanism for retrial customers. Retrial attempts from the orbit succeed only when the number of busy servers is at or below specific thresholds, which vary depending on the state of the arrival process. Customers waiting to retry may abandon the system independently due to impatience. When all servers become idle, they first remain on standby for a limited time, serving any arriving primary or retrial customers immediately; if no arrival occurs before this period ends, the servers take a synchronous working vacation, providing service at a reduced rate. Upon batch arrival, if insufficient servers are free, the batch is rejected or served partially, with the rest going into the orbit–a policy applied in both normal and working vacation modes. While in normal service mode, a disaster may occur at any time, causing all main servers to fail and forcing all customers in service to leave the system, causing no impact on the orbit. Repair begins immediately, during which no service is provided, and arriving batches may leave the system or join the orbit. The system state evolves as a multidimensional Markov chain in the asymptotically quasi-Toeplitz class, which facilitates obtaining the ergodicity conditions, enabling the derivation of steady-state probabilities and system performance measures. A case study involving the resolution of an optimization challenge is shown. Results highlight the influence of batch arrivals and system parameters on performance measures.

Keywords Batch Markovian arrival process · Phase-type service time · Customers impatience · Flexible retrial control · Synchronous working vacation · Disaster · Breakdown

Mathematics Subject Classification (2010) 60K25 · 60K30 · 68M20 · 90B22

✉ Agassi Melikov
amelikov@beu.edu.az

Natarajan Aishwarya
2401714001@ptuniv.edu.in

Govindan Ayyappan
ayyappan@ptuniv.edu.in

[1] Department of Mathematics, Puducherry Technological University, Puducherry, India

[2] Department of Mathematics, Baku Engineering University, Khirdalan, Azerbaijan

Published online: 04 February 2026

1 Introduction

In real-world systems, request retrials are a common occurrence, where an incoming request sees that all servers are busy; it leaves the service area but will come back to make the request again after a random amount of time. Consequently, this creates a complex scenario where repeated service attempts from a pool of repeated customers, termed the orbit, are combined with the standard flow of new arrivals. The monographs (Artalejo et al. 2008; Yang and Templeton 1989; Falin 1990; Falin and Templeton 1997) provide descriptions of the techniques for analyzing retrial queues as well as examples of such systems. Even though retrial queues are more mathematically challenging to study than queues with losses and buffers, they are a popular research topic because of their broad applicability in analyzing telecommunication networks and cloud computing. Furthermore, some customers might be patient and persistently retry to access the service, while others demonstrate impatience by abandoning the system after waiting for some time in orbit, a behavior often modeled as a competing risk between retrial and abandonment. This combination of multi-server queues with customer retrials and impatience, which accurately captures the behavior of real-world user request patterns, is precisely why this area has gained significant and sustained attention in queueing literature. Thus, this paper examines a multi-server retrial queueing model incorporating customer impatience.

Even the more sophisticated segments of the queueing theory literature on multi-server retrial queues generally assume that the arrival process is a stationary Poisson process with a fixed rate of arrivals. Although it is mathematically convenient, it often overlooks the non-stationary, correlated, and bursty nature of arrivals in practice. Markov arrival processes (*MAP*s) and batch Markov arrival processes (*BMAP*s) (Neuts 1979; Lucantoni 1991) have been developed as more flexible arrival models to address these limitations. Furthermore, phase-type (*PH*) distributions are considered advantageous in service operations due to their ability to accurately model diverse service time behaviors while simplifying mathematical analysis (Latouche and Ramaswami 1999). For instance, Breuer et al. (2002) explored the *BMAP*/*PH*/*N* retrial queue using a discrete-time, multidimensional, asymptotically quasi-Toeplitz Markov chain. Klimenok et al. (2007) investigated the *BMAP*/PH/N retrial queue with customer impatience. Kim et al. (2008) examined this system with a Markovian breakdown process, and later, in a Markovian random environment (Kim et al. 2009). The paper (Kim et al. 2010) analyzes tandem retrial queues with *BMAP* input, general service times, *PH*-type second phase, and customer loss, studying Markov chains, system states, and probabilities with numerical examples.

Although earlier work has extended the theory of *BMAP* retrial queues, a lot of it still assumes the simplifying restriction that new arrivals and retrial customers have equal probabilities of accessing idle servers. Such an assumption, though convenient analytically, may not correspond well with real-world service settings. A pragmatic approach is to differentiate service access between initial and re-attempting customers, as initial arrivals frequently constitute the most tangible metric of service in reality. Prioritizing primary arrivals decreases their blocking probability, thus increasing customer satisfaction, and retrial customers, who are more persistent, ensure eventual service utilization and thus contributing system profitability. Preferential treatment for main customers can be achieved by allocating a small number of servers exclusively for their use when few servers are available. A convenient way to implement this differentiation is through the use of threshold-based policies, in which a retrying customer is accepted only if the number of busy servers is less than a predetermined threshold.

This approach is validated by Dudin et al. (2024) in their analysis of the *MAP*/*M*/*N* retrial queue, where orbiting customers are accepted depending on the number of servers in use and on the state of the background arrival process. By incorporating state-dependent thresholds, the paper demonstrates how retrial attempts can be controlled: customers are served only if server occupancy is beneath the threshold; otherwise, they re-enter the orbit. The emergent dynamics are then modeled with a 3-dimensional Markov chain, and the stationary distribution yields performance information about the effectiveness of such controls. Recent work by Dudin et al. (2025) uses the *BMAP* for arrivals and *PH* service times. As with those models, it studies multi-server retrial queues with orbiting customers admitted through threshold-type control, where the system is modeled by a multidimensional Markov chain and performance is analyzed through steady-state analysis.

In real-world scenarios, servers occasionally like to take a break, handle other business, modeled as vacations in queueing systems. On vacations, the servers operate at a slow rate to prevent job accumulation and maintaining service. Over the past several years, a substantial body of literature has explored the retrial queueing model incorporating working vacations, as evidenced by studies (Ayyappan and Gowthami 2021; GnanaSekar and Kandaiyan 2022; Gupta and Kumar 2021; Li et al. 2018; Ayyappan and Thilagavathy 2024). Multi-server vacation queueing models with synchronous (group) or asynchronous (individual) vacations are less explored in retrial queues. Ke et al. (2024) studied a multi-server retrial model with synchronous working vacations, vacation interruption, impatience and constant retrial policy. Liu et al. (2024) investigated a multi-server two-way communication retrial queue with synchronous working vacation and a constant retrial policy. Subramanian et al. (2011) analyzed a multi-server retrial model with vacation policies, namely exhaustive service type vacation and Bernoulli vacation.

Recent studies have delved deeply into retrial queueing systems disrupted by catastrophic events or disasters. These disaster situation cause significant risks to the system's stability. When a disaster occurs, it triggers the immediate exit of one or more regular customers in service and causes a temporary server failure, rendering the service channel inoperative for a brief period. For an extensive discussion of disaster in retrial queues, see (Ammar and Rajadurai 2019; Li and Li 2020; Gao et al. 2021; Dimitriou 2013; Lisovskaya et al. 2022).

Extending the prior research on the multi-server retrial queueing model, this research proposes a novel model for cloud data center that integrates *BMAP* for arrivals, *PH*-distribution for service times, threshold-controlled retrials, synchronous working vacation and disaster-induced server breakdowns. Unlike earlier studies, our model captures the real-world complexity of such systems by incorporating bursty and correlated arrivals to model the bursty nature of user requests or data processing jobs arriving in batches, flexible service time distribution to represent the virtual machines (servers) with highly variable execution times for different tasks, prioritized access for primary customers through threshold-based policies to spin up a VM to handle backlogged requests (from the orbit) while conserving resources for primary arrivals, vacations of servers, and the impact of disasters (a host or rack failure) that disrupt service, causing all running tasks to be lost and requiring a recovery period. By parameterizing these characteristics within a single multidimensional Markov chain model and analyzing its steady-state behavior, this paper provides practical recommendations for system design and optimization, enhancing both efficiency and customer satisfaction. The theoretical framework is supported by numerical results to validate its effectiveness and applicability.

This paper is organized briefly as follows. Section 2 provides a brief mathematical description of the model we are addressing. Section 3 presents the modeling of the system states as a multidimensional continuous-time Markov chain (CTMC). We derive the infinitesimal generator for this chain in Section 4. The ergodicity condition for this chain is provided in this section. The system's performance measures are defined in Section 5. Numerical examples in Section 6 illustrate the effect of system parameters on performance metrics, and Section 7 concludes the paper.

2 Mathematical Model Description

Consider a multi-server retrial queueing model where customer arrivals to the system follow a *BMAP*. The arrival process is governed by a core stochastic process, ξ_t (for $t \geq 0$), modeled as an irreducible continuous-time Markov chain. This chain operates over a finite set of states labeled $\{1, 2, \ldots, m\}$ defined by the matrix representation $\{D_k, k = 1, 2, \ldots, K\}$ with an order of *m*, which capture the rates of transitions in the process ξ_t linked to the arrival of a group of *k* customers. Here, *K* indicates the largest possible batch size. Let D be the total sum of the batch arrivals, represented as $D = \sum_{k=1}^{K} D_k$, and let ζ denote the invariant vector corresponding to the irreducible generator $\tilde{D} = D_0 + D$. The vector ζ satisfies the conditions:

$$\zeta \tilde{D} = 0 \quad \text{and} \quad \zeta \mathrm{e} = 1.$$

The batch arrival rate, λ_g, and the average arrival rate, λ, can then be expressed as:

$$\lambda_g = \zeta D e \quad \text{and} \quad \lambda = \zeta \sum_{k=1}^{K} k D_k \mathrm{e}.$$

This system consists of *c* homogeneous servers without a waiting buffer. It is assumed that the service times follow a *PH* distribution with the representation $(\boldsymbol{\alpha}, T)$ of order *M*, where the average service rate is calculated as $\mu = [\boldsymbol{\alpha}(-T)^{-1}\mathrm{e}]^{-1}$. Instead of taking a vacation right away, servers go through a waiting period whose duration is exponentially distributed with parameter ν each time all servers become available. If a batch arrives or a retrial occurs during this waiting period, the servers provide service at the normal rate. However, if the waiting period ends without any arrivals, all servers simultaneously begin a synchronized working vacation, which lasts for an exponentially distributed time with parameter ω; during this vacation, they still serve customers but only at the reduced rate $(\boldsymbol{\alpha}', \phi T)$, where $0 < \phi < 1$.

When a group of customers arrives and finds enough idle servers available, they start processing on those servers. If there are not enough idle servers to accommodate the entire incoming batch, that batch faces rejection and leaves the system permanently with a probability of b_n (when the servers are in normal mode) and b_v (when the servers are in working vacation mode) or with the corresponding complementary probability, some of the batch will utilize all available idle servers, while the remaining customers will enter the orbit where they make individual and independent repeated attempts to access the service. The capacity of the orbit is considered to be infinite. The times between retries are modeled by an exponential distribution with the parameter θ, where $\theta > 0$. When there are i customers present in the orbit, the overall retrial rate equals θ_i.

In traditional retrial queues, a retrial is successful if there is at least one idle server available. In this context, we adopt a flexible retrial approach where a customer in the orbit can only receive service if, at the moment of retrial, the number of occupied servers does not surpass a certain integer threshold R_ξ, $0 \leq R_\xi < c$, $\xi = 1, 2, \dots, m$, which depends on the current state ξ of the underlying *BMAP* arrival process. If this condition is not met, the customer remains in the orbit.

Customers in orbit might leave the system independently without getting service because of impatience, where their time spent in orbit following an exponential distribution characterized by a rate $\gamma > 0$.

When the servers are busy in normal mode, a disaster can occur at any moment. The occurrence of disaster follows an exponential distribution with a rate of ψ. Upon the occurrence of a disaster, all the servers become non-functional and all customers in service are abruptly terminated from the system; however, the customers in the orbit remain unaffected. The repair process starts right away for all the servers to restore normal operation following a disaster event which lasts for an exponentially distributed time with parameter η. The probability of an arriving batch abandon the system when the servers are in repair is b_r. The process flow is illustrated in Fig. 1.

In a multi-server retrial system with s occupied servers and a service time distribution that follows PH and consists of M transient states, the state space of the multidimensional Markov chain describing the service process on occupied servers depends on the chosen tracking method.

The first approach, called track-phase-for-server (TPFS) in He and Alfa (2018), records the current service phase of each busy server individually. The second method, known as count-server-for-phase (CSFP) in He and Alfa (2018), instead counts the number of servers active in each service phase.

The CSFP method leads to a much smaller state space, with a size of: $d_s = \begin{pmatrix} s+M-1 \\ M-1 \end{pmatrix}$. For example, when $M = 2$ and $s = 15$, this results in only $s + 1 = 16$ states, whereas the TPFS method generates a state space of size M^s, which would be $2^{15} = 32,768$ states for the same parameters. Due to its smaller state space, the CSFP method offers greater computational efficiency and is therefore preferred for this model.

From this point forward, **0** represents a row vector filled with zeros, O signifies a zero matrix, I refers to an identity matrix, e stands for a unit column vector, all pertaining to their specific dimensions.

3 The Process of System States

Let

$$\boldsymbol{\Theta}_t = \{q_t, \xi_t, s_t, \tau_t, \Phi_t^{(1)}, \dots, \Phi_t^{(M)}\}, t \geq 0$$

be an irreducible CTMC describing the system, where the state at time *t* comprises:

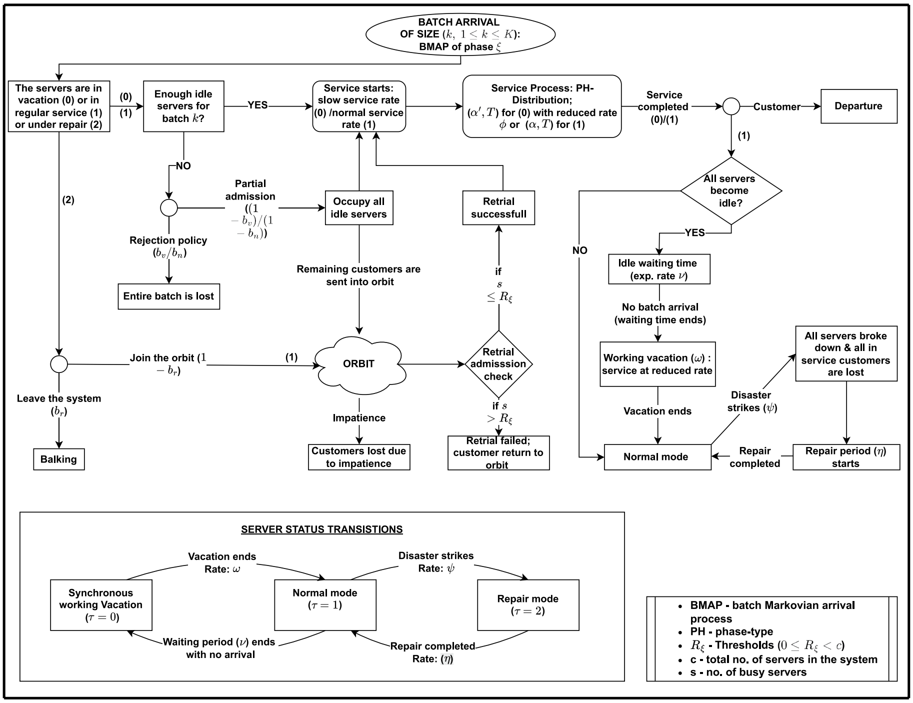

BATCH ARRIVAL OF SIZE (k, $1 < k < K$): BMAP of phase ξ
The servers are in vacation (0) or in regular service (1) or under repair (2)
(0)
(1)
Enough idle servers for batch k?
YES
Service starts: slow service rate (0) /normal service rate (1)
Service Process: PH-Distribution; (α', T) for (0) with reduced rate ϕ or (α, T) for (1)
Service completed (0)/(1)
Customer
Departure
(1)
All servers become idle?
NO
YES
NO
(2)
Partial admission ($(1-b_v)/(1-b_n)$)
Occupy all idle servers
Retrial successfull
Rejection policy (b_v/b_n)
Entire batch is lost
Remaining customers are sent into orbit
if $s \le R_\xi$
Idle waiting time (exp. rate ν)
No batch arrival (waiting time ends)
All servers broke down & all in service customers are lost
Join the orbit ($1-b_r$)
(1)
ORBIT
Retrial admisssion check
Working vacation (ω) : service at reduced rate
Disaster strikes (ψ)
Leave the system (b_r)
Impatience
if $s > R_\xi$
Vacation ends
Balking
Customers lost due to impatience
Retrial failed; customer return to orbit
Normal mode
Repair completed
Repair period (η) starts
SERVER STATUS TRANSISTIONS
Vacation ends Rate: ω
Disaster strikes Rate: ψ
Synchronous working Vacation ($\tau = 0$)
Normal mode ($\tau = 1$)
Repair mode ($\tau = 2$)
Waiting period (ν) ends with no arrival
Repair completed Rate: (η)
BMAP - batch Markovian arrival process
PH - phase-type
R_ξ - Thresholds ($0 \le R_\xi < c$)
c - total no. of servers in the system
s - no. of busy servers

Fig. 1 Schematic diagram illustrating the conceptual framework of the proposed model

- q_t = count of customers residing in the orbit, $q_t \ge 0$;
- ξ_t = phase of the *BMAP* arrival phase, $\xi_t = \overline{1,m}$;
- s_t = number of servers actively serving customers, $s_t = \overline{0,c}$;
- τ_t = server status, where

$$\tau_t = \begin{cases} 0, & \text{all servers are in vacation mode,} \\ 1, & \text{all servers are in normal service mode,} \\ 2, & \text{all servers are in repair period;} \end{cases}$$

- $\Phi_t^{(i)}$ = number of occupied servers in service phase i, $i = \overline{1,M}$, $\Phi_t^{(i)} = \overline{0,s_t}$, $\sum_{i=1}^{M} \Phi_t^{(i)} = s_t$.

The state space of this Markov chain is

$$\begin{aligned}\mathbf{\Theta}_t = &\{(q,\xi,s,\tau,\Phi^{(1)},\dots,\Phi^{(M)}) : q \ge 0, \xi = \overline{1,m}, s = \overline{0,c}, \tau = 0, \Phi^{(i)} = \overline{0,s}\} \\ &\cup \{(q,\xi,s,\tau,\Phi^{(1)},\dots,\Phi^{(M)}) : q \ge 0, \xi = \overline{1,m}, s = \overline{0,c}, \tau = 1, \Phi^{(i)} = \overline{0,s}\} \\ &\cup \{(q,\xi,s,\tau,\Phi^{(1)},\dots,\Phi^{(M)}) : q \ge 0, \xi = \overline{1,m}, s = 0, \tau = 2, \Phi^{(i)} = 0\}.\end{aligned}$$

We now present a few important matrices to characterize the auxiliary phase process $\Phi_t^{(i)}$, for $i = 1, 2, \dots, M$, which tracks the service phases across the c servers. Let s be the count of customers presently in service, with $s = 0, 1, \dots, c$. Following describes the changes: The matrix $P_s(\boldsymbol{\alpha})$ records the process's transition probabilities upon the arrival of a new customer starting service. The initial distribution $\boldsymbol{\alpha}$ determines how the system's state changes as the fresh customer joins one of the service phases. The square matrix $A_s(c,T)$represents the rates of transition of the process when an actively serving customer transitions from one service phase to another without finishing their current task. This matrix shows how the sub-generator matrix T controls the internal phase changes. The matrix L_{c-s} specifies the rates of transition for a service completion event, in which a customer completes service and leaves the system. This matrix explains why the number of busy servers has gone down and the service phases have changed to go along with it. To determine the overall exit rate from every state, we write the diagonal matrix Δ_s as:

$$\Delta_s = -\mathrm{diag}\left\{A_s(c,T)\mathrm{e} + L_{c-s}\mathrm{e}\right\}, \quad s = \overline{1,c}.$$

Here, Δ_s guarantees that the infinitesimal generator rows sum to zero by holding the negative sum of all outgoing transition rates from every state.

A comprehensive explanation of the matrices $P_s(\boldsymbol{\alpha})$, $A_s(c,T)$ and L_{c-s} along with the algorithms for the computation of these matrices is given in Kim et al. (2013).

4 The Generator Matrix

The Markov process $\boldsymbol{\Theta}_t, t \geq 0$ possesses an infinitesimal generator Q exhibiting an upper block-Hessenberg structure:

$$\mathrm{Q} = \begin{pmatrix} \mathrm{Q}_0^{(0)} & \mathrm{Q}_0^{(1)} & \mathrm{Q}_0^{(2)} & \mathrm{Q}_0^{(3)} & \cdots & \mathrm{Q}_0^{(K)} & O & O & O & \cdots \\ \mathrm{Q}_1^{(-1)} & \mathrm{Q}_1^{(0)} & \mathrm{Q}_1^{(2)} & \mathrm{Q}_1^{(3)} & \cdots & \mathrm{Q}_1^{(K)} & \mathrm{Q}_1^{(K+1)} & O & O & \cdots \\ O & \mathrm{Q}_2^{(-1)} & \mathrm{Q}_2^{(0)} & \mathrm{Q}_2^{(3)} & \cdots & \mathrm{Q}_2^{(K)} & \mathrm{Q}_2^{(K+1)} & \mathrm{Q}_2^{(K+2)} & O & \cdots \\ O & O & \mathrm{Q}_3^{(-1)} & \mathrm{Q}_3^{(0)} & \cdots & \mathrm{Q}_3^{(K)} & \mathrm{Q}_3^{(K+1)} & \mathrm{Q}_3^{(K+2)} & \mathrm{Q}_3^{(K+3)} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \quad (1)$$

Each block has dimension $m(3+2d) \times m(3+2d)$, where m is the number of *BMAP* phases, and $d = \sum_{s=1}^{c} \begin{pmatrix} s+M-1 \\ M-1 \end{pmatrix}$ is the total number of possible service phase distributions across all busy server counts. This captures $3m$ states when all servers are idle (with status vacation, normal) and under repair and the $2md$ states when servers are busy (with status vacation or normal, each with phase distribution).

The blocks are defined as follows:

1. Diagonal Blocks $\mathrm{Q}_q^{(0)}$ represent transitions within the same orbit level q, meaning no change in the number of customers in the orbit. These blocks are defined as follows:

$$\mathrm{Q}_q^{(0)} = \left(\mathrm{Q}_q^{(0)}\right)_{\xi}^{\xi'}, \quad q \geq 0, \quad \xi, \xi' = 1, \ldots, m,$$

where the sub-blocks are given by:

$$\left(\mathrm{Q}_q^{(0)}\right)_{\xi}^{\xi'} = \begin{cases} (D_0)_{\xi}^{\xi} I_{3+2d} + U_{\xi} - q\gamma I_{3+2d} + \theta_q \Gamma_{\xi}, & \xi = 1, \ldots, m, \\ \tilde{U}_{\xi,\xi'} + (D_0)_{\xi}^{\xi'} I_{3+2d}, & \xi, \xi' = 1, \ldots, m, \quad \xi \neq \xi'. \end{cases}$$

The matrix Γ_{ξ} is a diagonal matrix of size $(3+2d) \times (3+2d)$, defined as:

$$\Gamma_{\xi} = \mathrm{diag}\{0, \ldots, 0, 1, \ldots, 1\},$$

where zeros correspond to states with $s_t \leq R_{\xi}$, and ones correspond to states with $s_t > R_{\xi}$. The matrix U_{ξ} is a block upper-Hessenberg matrix of size $(3+2d) \times (3+2d)$, structured as $U_{\xi} = \left(U_{s,s'}^{\xi}\right)_{s,s'=0,\ldots,c}$, with non-zero blocks defined as follows:

$$U_{0,0}^{\xi} = \begin{bmatrix} -(\omega+\theta_q) + V_{0,0}^{\xi} & \omega & 0 \\ \nu & -(\nu+\theta_q) + V_{0,0}^{\xi} & 0 \\ 0 & \eta & b_r \left(\sum_{k=1}^{K} (D_k)_{\xi}^{\xi}\right) - \eta \end{bmatrix},$$

$$U_{s,s}^{\xi} = \begin{bmatrix} \phi(A_s(c,T) + \Delta_s) - \omega I_{d_s} + V_{s,s}^{\xi} & \omega I_{d_s} \\ 0 & (A_s(c,T) + \Delta_s) - \psi I_{d_s} + V_{s,s}^{\xi} \end{bmatrix} - \theta_q I_{2d_s}, \quad s = 1, \ldots, c,$$

where

$$V_{s,s}^{\xi} = \begin{cases} O_{d_s}, & 0 \leq s \leq c-K, \quad \tau = 0, 1, \\ b_v \sum_{k=c-s+1}^{K} (D_k)_{\xi}^{\xi} I_{d_s}, & c-K \leq s \leq c, \quad \tau = 0, \\ b_n \sum_{k=c-s+1}^{K} (D_k)_{\xi}^{\xi} I_{d_s}, & c-K \leq s \leq c, \quad \tau = 1. \end{cases}$$

Further, the off-diagonal blocks of U_{ξ} are:

$$U_{0,k}^{\xi} = \begin{bmatrix} (D_k)_{\xi}^{\xi} \prod_{u=0}^{k-1} P_u(\boldsymbol{\alpha}') & 0 \\ 0 & (D_k)_{\xi}^{\xi} \prod_{u=0}^{k-1} P_u(\boldsymbol{\alpha}) \\ 0 & 0 \end{bmatrix}, \quad k \leq c,$$

$$U_{s,s+k}^{\xi} = \begin{bmatrix} (D_k)_{\xi}^{\xi} \prod_{u=s}^{s+k-1} P_u(\boldsymbol{\alpha}') & 0 \\ 0 & (D_k)_{\xi}^{\xi} \prod_{u=s}^{s+k-1} P_u(\boldsymbol{\alpha}) \end{bmatrix}, \quad s+k \leq c, \quad k \leq K, \quad s = 1, \ldots, c-1,$$

$$U_{s,s-1}^{\xi} = \begin{bmatrix} \phi L_{c-s} & 0 \\ 0 & L_{c-s} \end{bmatrix}, \quad s = 2, \ldots, c,$$

$$U_{1,0}^{\xi} = \begin{bmatrix} \phi L_{c-1} & 0 & 0 \\ 0 & L_{c-1} & \psi e_{d_1} \end{bmatrix},$$

$$U_{s,0}^{\xi} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \psi e_{d_s} \end{bmatrix}, \quad s = 2, \ldots, c.$$

The matrix $\tilde{U}_{\xi,\xi'}$ is a block matrix of size $(3+2d) \times (3+2d)$, structured as $\tilde{U}_{\xi,\xi'} = \left(\tilde{U}_{s,s'}^{\xi,\xi'}\right)_{s,s'=0,\ldots,c}$, with non-zero blocks defined as follows:

$$\tilde{U}_{0,0}^{\xi,\xi'} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & b_r \sum_{k=1}^{K} (D_k)_{\xi}^{\xi'} \end{bmatrix},$$

$$\tilde{U}_{s,s} = \begin{bmatrix} \tilde{V}_{s,s}^{\xi,\xi'} & 0 \\ 0 & \tilde{V}_{s,s}^{\xi,\xi'} \end{bmatrix}, \quad s = 1, \ldots, c,$$

where

$$\tilde{V}_{s,s}^{\xi,\xi'} = \begin{cases} O_{d_s}, & 1 \le s \le c-K, \quad \tau = 0,1, \\ b_v \sum_{k=c-s+1}^{K} (D_k)_{\xi}^{\xi'} I_{d_s}, & c-K \le s \le c, \quad \tau = 0, \\ b_n \sum_{k=c-s+1}^{K} (D_k)_{\xi}^{\xi'} I_{d_s}, & c-K \le s \le c, \quad \tau = 1. \end{cases}$$

The off-diagonal blocks of $\tilde{U}_{\xi,\xi'}$ are:

$$\tilde{U}_{0,k}^{\xi,\xi'} = \begin{bmatrix} (D_k)_{\xi}^{\xi'} \prod_{u=0}^{k-1} P_u(\boldsymbol{\alpha'}) & 0 \\ 0 & (D_k)_{\xi}^{\xi'} \prod_{u=0}^{k-1} P_u(\boldsymbol{\alpha}) \\ 0 & 0 \end{bmatrix}, \quad k \le c,$$

$$\tilde{U}_{s,s+k}^{\xi,\xi'} = \begin{bmatrix} (D_k)_{\xi}^{\xi'} \prod_{u=s}^{s+k-1} P_u(\boldsymbol{\alpha'}) & 0 \\ 0 & (D_k)_{\xi}^{\xi'} \prod_{u=s}^{s+k-1} P_u(\boldsymbol{\alpha}) \end{bmatrix}, \quad s+k \le c, \quad k \le K, \quad s = 1,\ldots,c-1.$$

2. Sub-Diagonal Blocks $\mathrm{Q}_q^{(-1)}$, $q \ge 1$ represent transitions that reduce the orbit size by one, typically due to customer impatience or successful retrials. These blocks are defined as follows:

$$\mathrm{Q}_q^{(-1)} = \mathrm{diag}\left(\left(\mathrm{Q}_q^{(-1)}\right)_{\xi}^{\xi}, \quad q \ge 1, \quad \xi = 1,\ldots,m\right),$$

where the diagonal elements are given by:

$$\left(\mathrm{Q}_q^{(-1)}\right)_{\xi}^{\xi} = q\gamma I_{3+2d} + \theta_q \bar{U}_{\xi}.$$

The matrix $\bar{U}_{\xi}$ is a block matrix of size $(3+2d) \times (3+2d)$, structured as $\bar{U}_{\xi} = \left(\bar{U}_{s,s'}^{\xi}\right)_{s,s'=0,\ldots,c}$, with non-zero blocks defined as follows:

- For the transition from state $s = 0$ to $s = 1$:

$$\bar{U}_{0,1}^{\xi} = \begin{bmatrix} P_0(\boldsymbol{\alpha'}) & 0 \\ 0 & P_0(\boldsymbol{\alpha}) \\ 0 & 0 \end{bmatrix}.$$

- For transitions from state s to $s+1$, where $s = 1,\ldots,R_{\xi}$:

$$\bar{U}_{s,s+1}^{\xi} = \begin{bmatrix} P_s(\boldsymbol{\alpha'}) & 0 \\ 0 & P_s(\boldsymbol{\alpha}) \end{bmatrix}.$$

3. Super-Diagonal Blocks $\mathrm{Q}_q^{(q+k)}$, $k = 1,\ldots,K$: These describe transitions increasing the orbit size by k due to batch arrivals where some customers join the orbit. These blocks are defined as follows:

$$\mathrm{Q}_q^{(q+k)} = \left(\mathrm{Q}_q^{(q+k)}\right)_{\xi}^{\xi'}, \quad q \ge 0, \quad \xi,\xi' = 1,\ldots,m, \quad k = 1,\ldots,K,$$

where the blocks $\left(\mathrm{Q}_q^{(q+k)}\right)_{\xi}^{\xi'}$ have the following structure:

$$\left(\mathrm{Q}_q^{(q+k)}\right)_\xi^{\xi'} = \begin{bmatrix} (W_{0,0})_\xi^{\xi'} & O & \cdots & O & (W_{0,k})_\xi^{\xi'} \\ O & O & \cdots & O & (W_{1,k})_\xi^{\xi'} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & \cdots & O & (W_{c,k})_\xi^{\xi'} \end{bmatrix},$$

with the sub-matrices defined as:

$$(W_{0,0})_\xi^{\xi'} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & (1-b_r)(D_k)_\xi^{\xi'} \end{bmatrix}, \quad \xi, \xi' = 1, \ldots, m,$$

$$(W_{0,k})_\xi^{\xi'} = \begin{bmatrix} (\bar{W}_{s,k})_\xi^{\xi'} & 0 \\ 0 & (\bar{W}_{s,k})_\xi^{\xi'} \\ 0 & 0 \end{bmatrix},$$

$$(W_{s,k})_\xi^{\xi'} = \begin{bmatrix} (\bar{W}_{s,k})_\xi^{\xi'} & 0 \\ 0 & (\bar{W}_{s,k})_\xi^{\xi'} \end{bmatrix}, \quad s = 1, \ldots, c,$$

where the matrix $(\bar{W}_{s,k})_\xi^{\xi'}$ is given by:

$$(\bar{W}_{s,k})_\xi^{\xi'} = \begin{cases} O_{d_s}, & 0 \leq s < c - K + k, \quad \tau = 0, 1, \\ (1-b_v)(D_{c-s+k})_\xi^{\xi'} \prod_{w=s}^{c-1} P_s(\boldsymbol{\alpha'}), & c - K + k \leq s < c, \quad \tau = 0, \\ (1-b_n)(D_{c-s+k})_\xi^{\xi'} \prod_{w=s}^{c-1} P_s(\boldsymbol{\alpha}), & c - K + k \leq s < c, \quad \tau = 1, \\ (1-b_v)(D_k)_\xi^{\xi'} I_{d_c}, & s = c, \quad \tau = 0, \\ (1-b_n)(D_k)_\xi^{\xi'} I_{d_c}, & s = c, \quad \tau = 1. \end{cases}$$

As observed in Neuts (2021), the Markov chain $\boldsymbol{\Theta}_t$ is clearly not of the *M*/*G*/1 kind as the generator matrix blocks $\mathrm{Q}_q^{(0)}$ and $\mathrm{Q}_q^{(-1)}$ expressly rely on the level (q). This variation gives us significant challenges when trying to analyze characteristics of the chain. But one can prove the existence of limit matrices Y_f, for $f \geq 0$,

$$Y_0 = \lim_{q \to \infty} \tilde{R}_q^{-1} \mathbf{Q}_q^{(-1)}, \quad Y_1 = \lim_{q \to \infty} \tilde{R}_q^{-1} \mathbf{Q}_q^{(0)} + I, \quad Y_f = \lim_{q \to \infty} \tilde{R}_q^{-1} \mathbf{Q}_q^{(q+f-1)}, \ f \geq 2$$

where $\tilde{R}_q = -I \circ \mathrm{Q}_q^{(0)}$ where $\circ$ represents the entry-wise Hadamard product. The infinite sum $\sum_{f=0}^{\infty} Y_f$ forms a stochastic matrix, signifying that $\boldsymbol{\Theta}_t$ belongs to the class of Asymptotically Quasi-Toeplitz Markov Chain (AQTMC), defined in Klimenok and Dudin (2006). While utilizing the framework from Klimenok and Dudin (2006) typically demands computing the precise expressions for the Y_f matrices ($f \geq 0$), this step can be circumvented in our case, as a result of the assumption regarding customer impatience in the orbit ($\gamma > 0$). By drawing on the outcome from Dudin et al. (2024), we arrive at the subsequent conclusion.

Thus, for any combination of system parameters, the Markov chain $\boldsymbol{\Theta}_t$ is ergodic because of the impatience of customers remaining in the orbit ($\gamma > 0$). The long-term distribution of the chain is described by the ergodic character of $\{\boldsymbol{\Theta}_t, t \geq 0\}$, which ensures that its steady-state probabilities exist and are unique.

The stationary probabilities are defined as:

$$\boldsymbol{z}(q,\xi,s,\tau,\Phi^{(1)},\ldots,\Phi^{(M)}) = \lim_{t\to\infty} P\{q_t = q, \xi_t = \xi, s_t = s, \tau_t = \tau, \Phi_t^{(1)} = \Phi^{(1)}, \ldots, \Phi_t^{(M)} = \Phi^{(M)}\},$$

where: $q \geq 0,\ \xi = 1,\ldots,m,\ s = 0,\ldots,c,\ \tau = 0,1,2$ when $s = 0$, and $\tau = 0,1$ when $s > 0$, with $\sum_{i=1}^{M} \Phi^{(i)} = \overline{0,s}$.

To compute the stationary probabilities, we enumerate the states of $\boldsymbol{\Theta}_t$ in a lexicographic order:

Primary ordering: By q_t (orbit size, level q).

Secondary ordering: By ξ_t, s_t, τ_t, and the service phase counts $\{\Phi_t^{(1)}, \ldots, \Phi_t^{(M)}\}$. For the service phase counts, we use reverse lexicographic order to describe the states $\{\Phi^{(1)}, \ldots, \Phi^{(M)}\}$.

Let $\boldsymbol{z}_q$ denote the row vector of stationary probabilities for all states at level q. This vector is partitioned as:

$$\boldsymbol{z}_q = (\boldsymbol{z}(q,1), \boldsymbol{z}(q,2), \ldots, \boldsymbol{z}(q,m)),$$

within each $\boldsymbol{z}(q,\xi)$, we further partition by the number of busy servers s and server status τ:

$$\boldsymbol{z}(q,\xi) = (\boldsymbol{z}(q,\xi,0), \boldsymbol{z}(q,\xi,1), \ldots, \boldsymbol{z}(q,\xi,c)),$$

where for $s = 0$, $\boldsymbol{z}(q,\xi,0) = (\boldsymbol{z}(q,\xi,0,0), \boldsymbol{z}(q,\xi,0,1), \boldsymbol{z}(q,\xi,0,2))$, corresponding to $\tau = 0,1,2$, with dimension 3 (since $d_0 = 1$). For $s = 1,\ldots,c$, $\boldsymbol{z}(q,\xi,s) = (\boldsymbol{z}(q,\xi,s,0), \boldsymbol{z}(q,\xi,s,1))$, corresponding to $\tau = 0,1$, with each $\boldsymbol{z}(q,\xi,s,\tau)$ being a vector of dimension $d_s = \begin{pmatrix} s+M-1 \\ M-1 \end{pmatrix}$, covering the service phase counts $\{\Phi^{(1)}, \ldots, \Phi^{(M)}\}$.

Thus, the dimension of $\boldsymbol{z}(q,\xi)$ is: $3 + 2\sum_{s=1}^{c} d_s = 3 + 2d$, and the total dimension of $\boldsymbol{z}_q$ is: $m \cdot (3 + 2d)$.

The stationary probability vectors $\boldsymbol{z}_q, q \geq 0$, satisfy the system of linear algebraic equations:

$$(\boldsymbol{z}_0, \boldsymbol{z}_1, \boldsymbol{z}_2, \ldots)\mathrm{Q} = 0, \quad (\boldsymbol{z}_0, \boldsymbol{z}_1, \boldsymbol{z}_2, \ldots)\mathrm{e} = 1.$$

To compute the stationary probabilities $\boldsymbol{z}_q$, we employ the numerically stable algorithm described in Dudin et al. (2020). This algorithm avoids the need for explicit analytical derivations of the matrices $Y_f, f \geq 0$, and achieves a higher convergence rate by computing the transition probabilities of the finite components of the chain $\boldsymbol{\Theta}_t$ during the first passage time from level $q+1$ to level q.

5 Performance Measures

We now define key performance measures in terms of these steady-state probabilities.

1. The mean number of customers in the orbit, $\mathrm{E}_{\text{orbit}}$, represents the expected quantity of customers present in the orbit at any given time in the steady state.

$$\mathrm{E}_{\text{orbit}} = \sum_{q=1}^{\infty} q \boldsymbol{z}_q \mathrm{e}_{m(3+2d)}.$$

This can be decomposed by server status $\tau \in \{0, 1, 2\}$ as:

$$\mathrm{E}_{\text{orbit}} = \mathrm{E}_{\text{orbit}}^{(0)} + \mathrm{E}_{\text{orbit}}^{(1)} + \mathrm{E}_{\text{orbit}}^{(2)},$$

where:

$$\mathrm{E}_{\text{orbit}}^{(0)} = \sum_{q=1}^{\infty} q \sum_{\xi=1}^{m} \sum_{s=0}^{c} \boldsymbol{z}(q, \xi, s, 0) \mathrm{e}_{d_s},$$

$$\mathrm{E}_{\text{orbit}}^{(1)} = \sum_{q=1}^{\infty} q \sum_{\xi=1}^{m} \sum_{s=0}^{c} \boldsymbol{z}(q, \xi, s, 1) \mathrm{e}_{d_s},$$

$$\mathrm{E}_{\text{orbit}}^{(2)} = \sum_{q=1}^{\infty} q \sum_{\xi=1}^{m} \boldsymbol{z}(q, \xi, 0, 2).$$

2. The mean number of busy servers, c_{busy}, represents the expected number of servers occupied by customers at any given time in the steady state. It is given by:

$$\mathrm{c}_{\text{busy}} = \sum_{q=0}^{\infty} \sum_{\xi=1}^{m} \sum_{s=1}^{c} s \sum_{\tau=0}^{1} \boldsymbol{z}(q, \xi, s, \tau) \mathrm{e}_{d_s},$$

and can be decomposed by server status $\tau \in \{0, 1\}$ as:

$$\mathrm{c}_{\text{busy}} = \mathrm{c}_{\text{busy}}^{(0)} + \mathrm{c}_{\text{busy}}^{(1)},$$

where:

$$\mathrm{c}_{\text{busy}}^{(0)} = \sum_{q=0}^{\infty} \sum_{\xi=1}^{m} \sum_{s=1}^{c} s \boldsymbol{z}(q, \xi, s, 0) \mathrm{e}_{d_s},$$

$$\mathrm{c}_{\text{busy}}^{(1)} = \sum_{q=0}^{\infty} \sum_{\xi=1}^{m} \sum_{s=1}^{c} s \boldsymbol{z}(q, \xi, s, 1) \mathrm{e}_{d_s}.$$

3. The mean number of customers in the system, $\mathrm{E}_{\text{system}}$, represents the expected total number of customers in the orbit and in service at any given time in the steady state. It is given by:

$$\mathrm{E}_{\text{system}} = \sum_{q=0}^{\infty}\sum_{\xi=1}^{m}\left(\sum_{s=0}^{c}(q+s)\sum_{\tau=0}^{1}\boldsymbol{z}(q,\xi,s,\tau)\mathrm{e}_{d_s} + q\boldsymbol{z}(q,\xi,0,2)\right),$$

which is equivalent to:

$$\mathrm{E}_{\text{system}} = \mathrm{E}_{\text{orbit}} + \mathrm{c}_{\text{busy}},$$

and can be decomposed by server status $\tau \in \{0,1,2\}$ as:

$$\mathrm{E}_{\text{system}} = \mathrm{E}^{(0)}_{\text{system}} + \mathrm{E}^{(1)}_{\text{system}} + \mathrm{E}^{(2)}_{\text{system}},$$

where:

$$\mathrm{E}^{(0)}_{\text{system}} = \sum_{q=0}^{\infty}\sum_{\xi=1}^{m}\sum_{s=0}^{c}(q+s)\boldsymbol{z}(q,\xi,s,0)\mathrm{e}_{d_s},$$

$$\mathrm{E}^{(1)}_{\text{system}} = \sum_{q=0}^{\infty}\sum_{\xi=1}^{m}\sum_{s=0}^{c}(q+s)\boldsymbol{z}(q,\xi,s,1)\mathrm{e}_{d_s},$$

$$\mathrm{E}^{(2)}_{\text{system}} = \sum_{q=1}^{\infty}q\sum_{\xi=1}^{m}\boldsymbol{z}(q,\xi,0,2).$$

4. The probability that all servers are idle, $\mathrm{P}_{\text{c-idle}}$, represents the probability that no servers are occupied by customers ($s = 0$) and the servers are in either vacation mode ($\tau = 0$) or normal mode ($\tau = 1$) at an arbitrary moment in the steady state. It is given by:

$$\mathrm{P}_{\text{c-idle}} = \sum_{q=0}^{\infty}\sum_{\xi=1}^{m}\sum_{\tau=0}^{1}\boldsymbol{z}(q,\xi,0,\tau),$$

and can be decomposed by server status $\tau \in \{0,1\}$ as:

$$\mathrm{P}_{\text{c-idle}} = \mathrm{P}^{(0)}_{\text{c-idle}} + \mathrm{P}^{(1)}_{\text{c-idle}},$$

where:

$$\mathrm{P}^{(0)}_{\text{c-idle}} = \sum_{q=0}^{\infty}\sum_{\xi=1}^{m}\boldsymbol{z}(q,\xi,0,0),$$

$$\mathrm{P}^{(1)}_{\text{c-idle}} = \sum_{q=0}^{\infty}\sum_{\xi=1}^{m} \boldsymbol{z}(q,\xi,0,1).$$

5. The probability that the orbit is empty, $\mathrm{P}_{\text{q-empty}}$, represents the probability that there are no customers in the orbit at an arbitrary moment in the steady state. It is given by:

$$\mathrm{P}_{\text{q-empty}} = \boldsymbol{z}_0 \mathrm{e}_{m(3+2d)} = \sum_{\xi=1}^{m}\sum_{s=0}^{c}\sum_{\tau\in T_s} \boldsymbol{z}(0,\xi,s,\tau)\mathrm{e}_{d_s},$$

where $T_s = \{0,1,2\}$ for $s = 0$, and $T_s = \{0,1\}$ for $s \geq 1$. This can be decomposed by server status $\tau \in \{0,1,2\}$ as:

$$\mathrm{P}_{\text{q-empty}} = \mathrm{P}^{(0)}_{\text{q-empty}} + \mathrm{P}^{(1)}_{\text{q-empty}} + \mathrm{P}^{(2)}_{\text{q-empty}},$$

where:

$$\mathrm{P}^{(0)}_{\text{q-empty}} = \sum_{\xi=1}^{m}\sum_{s=0}^{c} \boldsymbol{z}(0,\xi,s,0)\mathrm{e}_{d_s},$$

$$\mathrm{P}^{(1)}_{\text{q-empty}} = \sum_{\xi=1}^{m}\sum_{s=0}^{c} \boldsymbol{z}(0,\xi,s,1)\mathrm{e}_{d_s},$$

$$\mathrm{P}^{(2)}_{\text{q-empty}} = \sum_{\xi=1}^{m} \boldsymbol{z}(0,\xi,0,2).$$

6. The probability that the system is empty, $\mathrm{P}_{\text{empty}}$, represents the probability that there are no customers in the orbit or in service at an arbitrary moment in the steady state. It is given by:

$$\mathrm{P}_{\text{empty}} = \sum_{\xi=1}^{m}\sum_{\tau=0}^{2} \boldsymbol{z}(0,\xi,0,\tau),$$

and can be decomposed by server status $\tau \in \{0,1,2\}$ as:

$$\mathrm{P}_{\text{empty}} = \mathrm{P}^{(0)}_{\text{empty}} + \mathrm{P}^{(1)}_{\text{empty}} + \mathrm{P}^{(2)}_{\text{empty}},$$

where:

$$\mathrm{P}^{(0)}_{\text{empty}} = \sum_{\xi=1}^{m} \boldsymbol{z}(0,\xi,0,0),$$

$$\mathrm{P}_{\text{empty}}^{(1)} = \sum_{\xi=1}^{m} \boldsymbol{z}(0,\xi,0,1),$$

$$\mathrm{P}_{\text{empty}}^{(2)} = \sum_{\xi=1}^{m} \boldsymbol{z}(0,\xi,0,2).$$

7. The probability that the system is in vacation mode ($\tau = 0$) is given by the sum of steady-state probabilities over all states where servers are in synchronous working vacation:

$$\mathrm{P}_{\text{ser-vac}} = \sum_{q=0}^{\infty}\sum_{\xi=1}^{m}\sum_{s=0}^{c} \mathrm{z}(q,\xi,s,0)\mathrm{e}_{d_s}.$$

8. The probability that the system is in repair mode ($\tau = 2$) is the sum of steady-state probabilities over all states where all servers are under repair:

$$\mathrm{P}_{\text{ser-repair}} = \sum_{q=0}^{\infty}\sum_{\xi=1}^{m} \mathrm{z}(q,\xi,0,2)\mathrm{e}.$$

9. The probability that the system is in normal mode ($\tau = 1$) is the sum of steady-state probabilities over all states where servers are operating at the normal service rate:

$$\mathrm{P}_{\text{ser-normal}} = \sum_{q=0}^{\infty}\sum_{\xi=1}^{m}\sum_{s=0}^{c} \mathrm{z}(q,\xi,s,1)\mathrm{e}_{d_s}.$$

10. The output flow intensity λ_{out}, representing the rate at which customers complete service and leave the system, is given by:

$$\lambda_{\text{out}} = \sum_{q=0}^{\infty}\sum_{\xi=1}^{m}\sum_{s=1}^{c} \left(\mathrm{z}(q,\xi,s,0)\phi L_{c-s}\mathrm{e}_{d_{s-1}} + \mathrm{z}(q,\xi,s,1)L_{c-s}\mathrm{e}_{d_{s-1}}\right).$$

11. The probability that an arbitrary customer abandons the orbit due to impatience, denoted by $\mathrm{P}_{\text{lost}}^{\text{imp}}$, is computed as follows:

$$\mathrm{P}_{\text{lost}}^{\text{imp}} = \frac{\gamma \cdot \mathrm{E}_{\text{orbit}}}{\lambda}.$$

The mean orbit size $\mathrm{E}_{\text{orbit}}$ is given by:

$$\mathrm{E}_{\text{orbit}} = \sum_{q=1}^{\infty} q \cdot \boldsymbol{z}_q \boldsymbol{e}.$$

Substituting the expression for $\mathrm{E}_{\text{orbit}}$, we obtain the final formula:

$$\mathrm{P}_{\text{lost}}^{\text{imp}} = \frac{\gamma}{\lambda}\sum_{q=1}^{\infty} q \cdot \boldsymbol{z}_q \boldsymbol{e}.$$

The probability $\mathrm{P}_{\text{lost}}^{(\xi),\text{imp}}$, representing the likelihood that a customer is lost from the orbit due to impatience when the $BMAP's$ underlying process is in state $\xi \in \{1, 2, \ldots, m\}$, is defined as follows:

$$\mathrm{P}_{\text{lost}}^{(\xi),\text{imp}} = \frac{\gamma}{\lambda}\left(\sum_{q=1}^{\infty}\sum_{s=0}^{c}\sum_{\tau=0}^{1} q\mathrm{z}(q, \xi, s, \tau)\mathrm{e}_{d_s} + \sum_{q=1}^{\infty} q\mathrm{z}(q, \xi, 0, 2)\right).$$

12. The probability $\mathrm{P}_{\text{lost}}^{\text{insuff}}$, which denotes the likelihood that an arbitrary customer is lost upon arrival due to insufficient idle servers is derived as follows:

$$\mathrm{P}_{\text{lost}}^{\text{insuff}} = \frac{1}{\lambda}\left\{\sum_{q=0}^{\infty}\sum_{\xi=1}^{m}\sum_{\xi'=1}^{m}\sum_{s=0}^{c}\sum_{k=c-s+1}^{K} k\left(b_v(D_k)_{\xi}^{\xi'}\mathrm{z}(q, \xi, s, 0)\mathrm{e}_{d_s} + b_n(D_k)_{\xi}^{\xi'}\mathrm{z}(q, \xi, s, 1)\mathrm{e}_{d_s}\right)\right\}.$$

The likelihood $\mathrm{P}_{\text{lost}}^{(\xi),\text{insuff}}$, that a customer is lost upon arrival due to an insufficient number of idle servers, given that the $BMAP's$ underlying process ξ_t is in state $\xi \in \{1, 2, \ldots, m\}$, is calculated as:

$$\mathrm{P}_{\text{lost}}^{(\xi),\text{insuff}} = \frac{1}{\lambda}\left\{\sum_{q=0}^{\infty}\sum_{\xi'=1}^{m}\sum_{s=0}^{c}\sum_{k=c-s+1}^{K} k\left(b_v(D_k)_{\xi}^{\xi'}\mathrm{z}(q, \xi, s, 0)\mathrm{e}_{d_s} + b_n(D_k)_{\xi}^{\xi'}\mathrm{z}(q, \xi, s, 1)\mathrm{e}_{d_s}\right)\right\}.$$

13. The probability P_{balk} accounts for losses when a batch of any size $k \geq 1$ arrives during repair mode ($\tau = 2$, $s = 0$), and the entire batch is lost with probability b_r.

$$\mathrm{P}_{\text{balk}} = \frac{1}{\lambda}\sum_{q=0}^{\infty}\sum_{\xi=1}^{m}\sum_{\xi'=1}^{m}\sum_{k=1}^{K} kb_r(D_k)_{\xi}^{\xi'}\mathrm{z}(q, \xi, 0, 2).$$

The probability $\mathrm{P}_{\text{balk}}^{(\xi)}$ accounts for losses when a batch of any size $k \geq 1$ arrives during repair mode ($\tau = 2$, $s = 0$), and the entire batch is lost with probability b_r, given that the *BMAP*'s underlying process ξ_t is in state $\xi \in \{1, 2, \ldots, m\}$, is given as:

$$\mathrm{P}_{\text{balk}}^{(\xi)} = \frac{1}{\lambda}\sum_{q=0}^{\infty}\sum_{\xi'=1}^{m}\sum_{k=1}^{K} kb_r(D_k)_{\xi}^{\xi'}\mathrm{z}(q, \xi, 0, 2).$$

14. Total Arrival Loss Probability

$$\mathrm{P}_{\text{lost}}^{\text{arrival}} = \mathrm{P}_{\text{lost}}^{\text{insuff}} + \mathrm{P}_{\text{balk}}.$$

15. The probability P_{lost} for a random customer is lost is determined by the formula

$$\mathrm{P}_{\text{lost}} = 1 - \frac{\lambda_{\text{out}}}{\lambda},$$

$$\mathrm{P}_{\text{lost}} = \mathrm{P}_{\text{lost}}^{\text{arrival}} + \mathrm{P}_{\text{lost}}^{\text{imp}}.$$

16. The probability $\mathrm{P}_{\text{direct-service}}$, which indicates the probability that a customer, immediately begins service without entering the orbit is given by:

$$\begin{aligned}
\mathrm{P}_{\text{direct-service}} = \frac{1}{\lambda}\Bigg\{\sum_{q=0}^{\infty}\sum_{\xi=1}^{m}\sum_{\xi'=1}^{m}\sum_{s=0}^{c-1}\Big(&\sum_{k=1}^{\min(K,c-s)} k\cdot(D_k)_{\xi}^{\xi'}\boldsymbol{z}(q,\xi,s,0)\boldsymbol{e} \\
&+ \sum_{k=1}^{\min(K,c-s)} k\cdot(D_k)_{\xi}^{\xi'}\boldsymbol{z}(q,\xi,s,1)\boldsymbol{e} \\
&+ \sum_{k=c-s+1}^{K}(1-b_v)(c-s)\cdot(D_k)_{\xi}^{\xi'}\boldsymbol{z}(q,\xi,s,0)\boldsymbol{e} \\
&+ \sum_{k=c-s+1}^{K}(1-b_n)(c-s)\cdot(D_k)_{\xi}^{\xi'}\boldsymbol{z}(q,\xi,s,1)\boldsymbol{e}\Big)\Bigg\}.
\end{aligned}$$

6 Numerical Results

This section examines the qualitative properties of performance metrics through several numerical analyzes.

Customers arrive in batches via a two-phase *BMAP* with a maximum batch size $K = 4$. The arrival dynamics are governed by the matrices

$$\begin{aligned}
D_0 &= \begin{pmatrix} -2.98717 & 0 \\ 0 & -6.12005 \end{pmatrix}, \\
D_1 &= \begin{pmatrix} 1.27501 & 0 \\ 0 & 2.91431 \end{pmatrix}, \quad D_2 = \begin{pmatrix} 0.72858 & 0 \\ 0 & 1.63930 \end{pmatrix}, \\
D_3 &= \begin{pmatrix} 0 & 0.54643 \\ 0.91072 & 0 \end{pmatrix}, \quad D_4 = \begin{pmatrix} 0 & 0.43715 \\ 0.65572 & 0 \end{pmatrix}.
\end{aligned}$$

The selected structure is designed so that the arrivals of small batches ($k = 1, 2$) occur without changing the underlying Markov phase, whereas larger batches ($k = 3, 4$) are necessarily accompanied by phase transitions. This modeling assumption captures practical situations in which sudden surges in customer demand represented by large group arrivals are accompanied by abrupt changes in the traffic regime, a phenomenon commonly observed in bursty service and communication systems.

The stationary distribution of the underlying Markov chain is $\boldsymbol{\zeta} = (0.6143,\ 0.3857)$, indicating unequal sojourn times across the two phases. The conditional mean arrival rates are $\lambda^{(1)} = 6.1201$ and $\lambda^{(2)} = 11.5480$ customers per unit time, indicating a clear distinction between a moderate and a high-arrival regime. The overall mean customer arrival rate is $\lambda = 8.2137$, while the mean batch arrival rate is $\lambda_g = 4.1956$.

Furthermore, the squared coefficient of variation of the inter-arrival times is $c_{\text{var}}^2 = 1.2544$ and the correlation coefficient between successive inter-arrival times is $c_{\text{cor}} = 0.0421$, confirming the presence of burstiness and positive correlation in the arrival stream. These characteristics make the selected *BMAP* particularly suitable for evaluating the performance of state-dependent retrial admission and threshold-based control policies.

Each customer's service time follows a phase-type distribution, defined by an initial probability vector $\boldsymbol{\alpha} = \boldsymbol{\alpha'} = (0.5, 0.5)$ and the matrix:

$$T = \begin{pmatrix} -2 & 0 \\ 0 & -2/3 \end{pmatrix},$$

such that $Te + T^0 = 0$. The average time to serve a customer is 1 time unit, with a squared coefficient of variation of 4, suggesting considerable variation in service times. The system operates with 14 servers.

The remaining parameters are fixed as follows: The rate at which customers in the orbit attempt to re-access the system is given by $\theta_q = q\theta$, where $q \geq 1$, having $\theta = 0.2$. Additionally, $\nu = 0.8,\ \omega = 0.3,\ \phi = 0.7,\ b_n = 0.4,\ b_v = 0.4,\ \gamma = 0.009,\ \psi = 0.1,\ \eta = 1$, and $b_r = 0.3$.

6.1 Illustration 1

The objective of this illustration is to quantify the impact of phase-dependent retrial thresholds on system congestion, customer losses, and profitability, as influenced by the control strategy parameters R_ξ, where ξ ranges from 1 to m, and to identify optimal threshold combinations under competing performance criteria.

We employ three-dimensional surface plots to visualize how key performance metrics change as we systematically adjust both thresholds R_1 and R_2 across their entire possible range (from 0 to $c-1$).

1. **Dependence of $\mathrm{E}_{\text{orbit}}$ on $\boldsymbol{R_1}$ and $\boldsymbol{R_2}$** Figure 2 shows the mean number of customers in the orbit, $\mathrm{E}_{\text{orbit}}$, as a function of the phase-dependent retrial thresholds (R_1, R_2). $\mathrm{E}_{\text{orbit}}$ decreases monotonically with increasing thresholds, from 73.41 under the most restrictive policy $(R_1, R_2) = (0, 0)$ to 5.18 under the fully permissive policy $(R_1, R_2) = (13, 13)$. A pronounced asymmetry in the influence of the two thresholds is observed. Increasing R_1 from 0 to 13 while keeping $R_2 = 0$ reduces $\mathrm{E}_{\text{orbit}}$ by approximately 89.4%, whereas increasing R_2 from 0 to 13 with fixed $R_1 = 0$ yields only an 83.6% reduction. This difference arises from the structure of the underlying *BMAP*. The system spends most of its time in the moderate-arrival phase (phase 1, stationary probability 0.614), whereas transitions to the high-arrival phase occur mainly due to large batch arrivals. As a result, allowing more retrial access during the dominant moderate-load phase is particularly effective in reducing orbit congestion.
2. **Dependence of $\mathrm{P}_{\text{lost}}^{\text{imp}}$ on $\boldsymbol{R_1}$ and $\boldsymbol{R_2}$** Figure 3 illustrates the impatience loss probability as a function of the phase-dependent retrial thresholds (R_1, R_2). Since the impatience loss is proportional to the mean orbit size, it decreases monotonically as either threshold increases. The highest loss probability, equal to 0.0805, is observed under the most

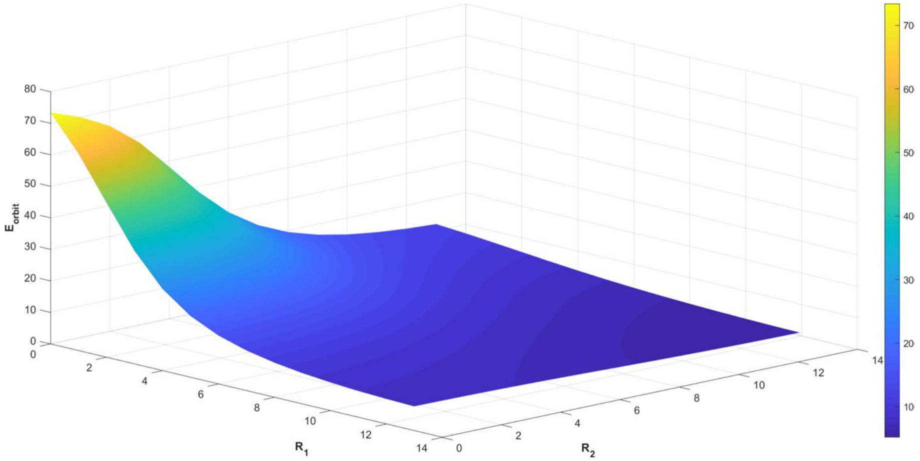


Fig. 2 Mean number of customers in the orbit ($\mathrm{E}_{\mathrm{orbit}}$) as a function of state-dependent thresholds R_1 and R_2

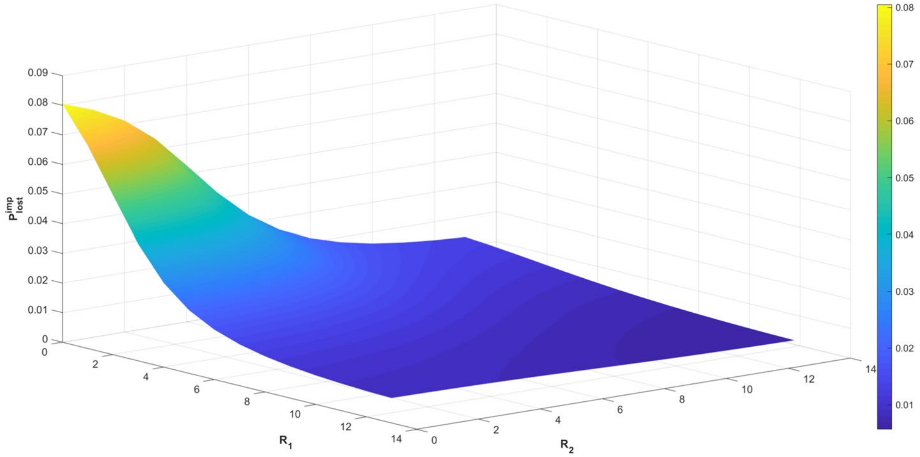


Fig. 3 Influence of R_1 and R_2 on impatience loss probability, $\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}0$

restrictive retrial policy $(R_1, R_2) = (0, 0)$, where customers experience long waiting times in the orbit. As the thresholds are relaxed, retrial customers gain more frequent access to service, leading to a steady reduction in orbit congestion and, consequently, in impatience-induced losses. Under the fully permissive policy $(R_1, R_2) = (13, 13)$, the impatience loss probability drops to 0.0057, indicating that only a small fraction of customers abandon the system due to excessive waiting in the orbit.

3. **Trade-off Between Retrial Access and Primary Customer Loss** Figure 4 presents the probability $\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$ that an arriving primary customer is lost due to insufficient idle servers, while Figs. 5 and 6 break it down by phase. In contrast to the decreasing trends observed for orbit size and impatience losses, $\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$ increases monotonically with higher thresholds, rising from 0.0351 under the restrictive policy (0, 0) to 0.0486

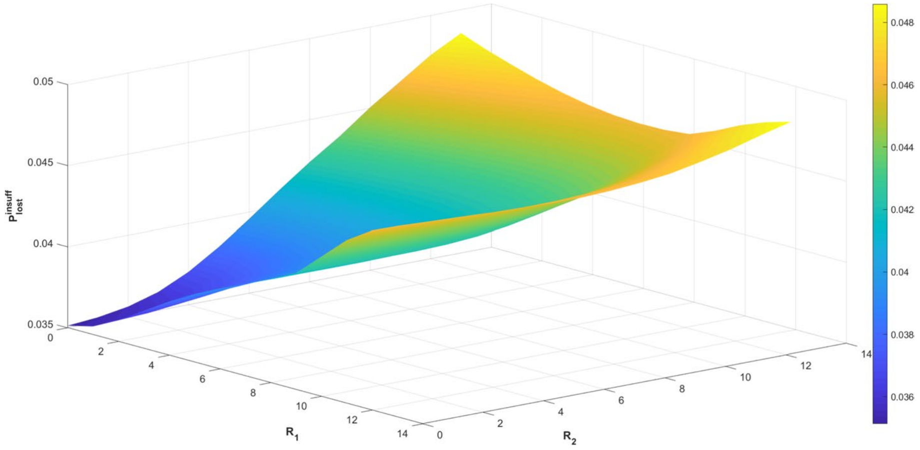


Fig. 4 Impact of R_1 and R_2 on loss probability due to insufficient servers, $\mathrm{P}_{\text{lost}}^{\text{insuff}}$

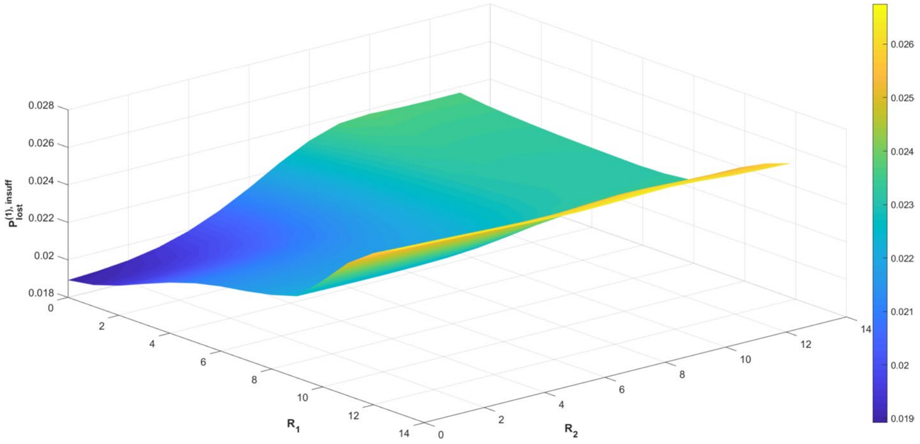


Fig. 5 Variation in state 1 primary customer loss probability, $\mathrm{P}_{\text{lost}}^{(1),\text{insuff}}$, with R_1 and R_2

under the permissive policy (13, 13). This reflects the inherent trade-off: more liberal retrial admission occupies additional servers, elevating the risk of batch rejection for primary arrivals. Notably, the threshold R_2, corresponding to the high-arrival burst phase, induces a stronger increase in losses. Liberalizing retrials solely in this phase (increasing R_2 from 0 to 13 with $R_1 = 0$) raises $\mathrm{P}_{\text{lost}}^{\text{insuff}}$ by 38.2%, compared to a 33.2% increase when liberalizing solely in the dominant moderate phase (R_1 from 0 to 13 with $R_2 = 0$). This asymmetric effect is consistent with the *BMAP*-induced phase behavior discussed earlier.

Overall, these results reveal a clear trade-off between retrial accessibility and primary customer protection: more permissive retrial thresholds substantially reduce orbit congestion and impatience losses, but at the cost of increased primary customer loss due to insuffi-

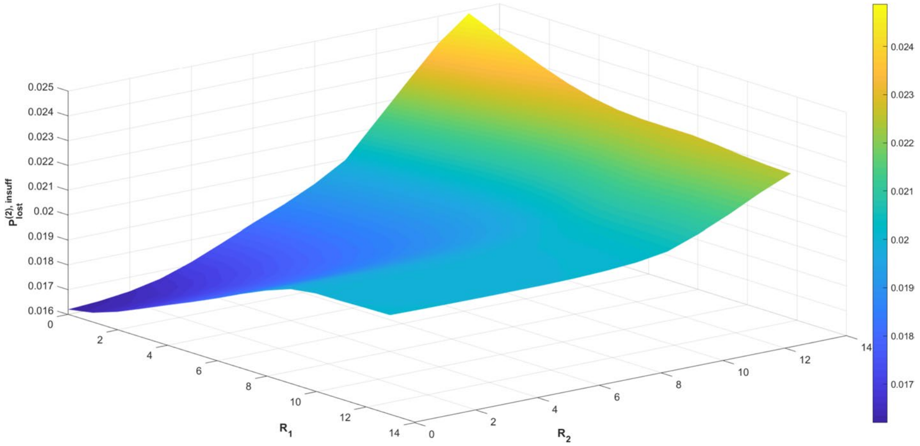


Fig. 6 Variation in state 2 primary customer loss probability, $\mathrm{P}^{(2),\mathrm{insuff}}_{\mathrm{lost}}$, with R_1 and R_2

cient server availability. Consistent with the asymmetric effects observed across phases, liberalizing the threshold in the moderate-arrival phase R_1 yields particularly favourable performance, achieving reductions of over 89% in orbit size and abandonment with only a moderate increase in primary losses (approximately 33%). In contrast, relaxing the threshold in the high-arrival burst phase R_2 provides smaller congestion relief while inducing a larger increase in primary losses (approximately 38%). Consequently, effective state-dependent retrial policies in bursty and correlated environments should favour relatively high values of R_1 and more conservative settings of R_2 to attain balanced and robust performance gains.

Optimization of Retrial Thresholds Figure 7 depicts the total customer loss probability ($\mathrm{P}_{\mathrm{lost}}$), which aggregates primary customer losses and losses due to customer impatience in the orbit, across the range of phase-dependent thresholds (R_1, R_2). The resulting loss surface exhibits a minimum at $(R_1, R_2) = (9, 8)$, where the total loss probability attains its minimum value of approximately 0.07818. This represents a substantial improvement of over 45% relative to the most restrictive policy (0, 0) and approximately 4% relative to the fully permissive policy (13, 13).

The optimal policy is asymmetric, employing a relatively liberal threshold in the moderate-arrival phase ($R_1 = 9$) and a slightly more conservative threshold in the high-arrival burst phase ($R_2 = 8$). Consistent with the phase-dependent effects observed earlier, this configuration effectively balances retrial access against server availability, thereby minimizing the overall loss probability.

Optimization of Thresholds to Maximize Average Profit To illustrate the practical implications of the threshold policy, we consider the net revenue function

$$C(R_1, R_2) = a_1\lambda_{\mathrm{out}} - a_2\lambda\mathrm{P}^{\mathrm{arrival}}_{\mathrm{lost}} - a_3\lambda\mathrm{P}^{\mathrm{imp}}_{\mathrm{lost}},$$

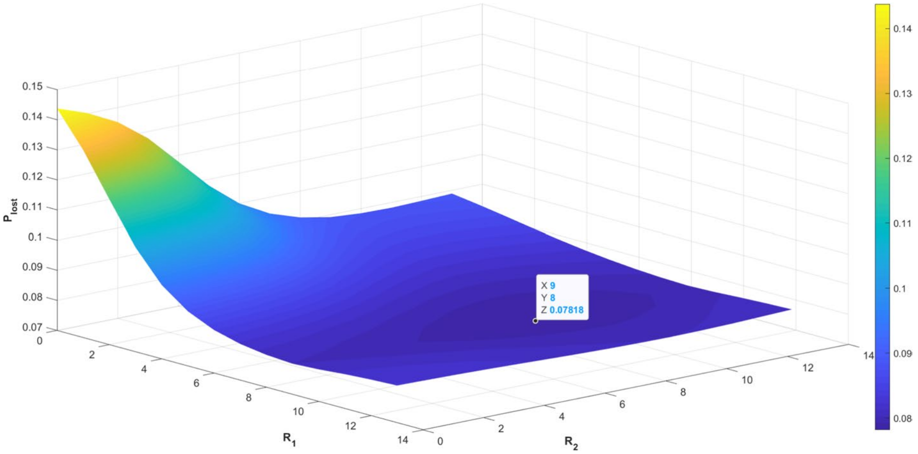


Fig. 7 Effect of R_1 and R_2 on total loss probability, $\mathrm{P}_{\mathrm{lost}}$

where λ_{out} is the effective output rate of successfully served customers, $\mathrm{P}_{\mathrm{lost}}^{\mathrm{arrival}}$ is the probability of primary customer loss, and $\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}$ is the loss probability due to customer impatience in the orbit. The coefficients a_1, a_2, and a_3 represent the revenue per served customer and costs per lost customer (primary and impatient), respectively. Unless stated otherwise, we set $a_1 = 1$, $a_2 = 5$, and $a_3 = 2$.

Figure 8 depicts the net revenue surface $C(R_1, R_2)$ over the threshold grid. The maximum net revenue is attained at $(R_1, R_2) = (8, 6)$ with $C \approx 4.57$, which is significantly higher than that achieved under the restrictive policy (0, 0) and also exceed the revenue obtained under fully permissive policy (13, 13). The optimal policy is asymmetric, employing a higher threshold in the moderate-arrival phase ($R_1 = 8$) and a lower threshold in the high-arrival burst phase ($R_2 = 6$). This configuration balances the benefits of orbit clearing against the risk of primary batch rejection, thereby maximizing overall system profitability.

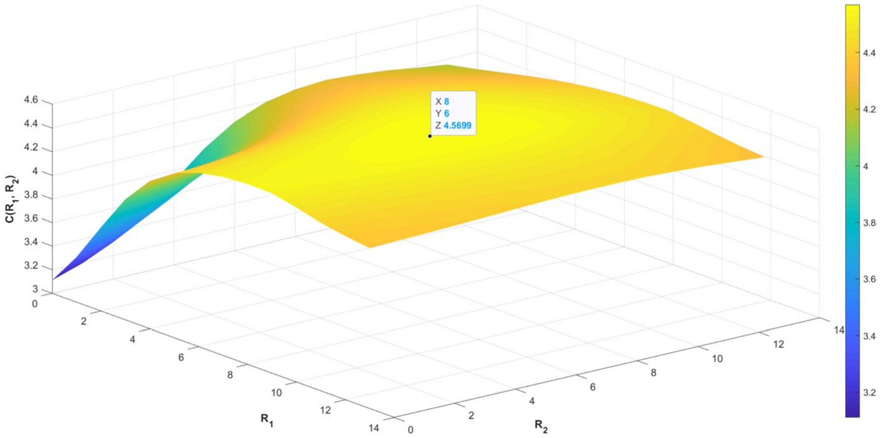


Fig. 8 Influence of R_1 and R_2 on cost function, $C(R_1, R_2)$

6.2 Illustration 2

To demonstrate the impact of explicitly modeling batch arrivals, we compare the system performance under the considered *BMAP* with that obtained under a matched *MAP* that preserves the same mean arrival rate, variability, and inter-arrival correlation. This comparison is intended to assess the extent to which ignoring batch arrivals affects performance evaluation and threshold optimization in bursty and correlated traffic environments.

The underlying matrices of this *MAP* are

$$D_0 = \begin{pmatrix} -5.848 & 0 \\ 0 & -11.9812 \end{pmatrix}, \quad D_1 = \begin{pmatrix} 3.9224 & 1.9256 \\ 3.0666 & 8.9146 \end{pmatrix},$$

yielding the arrival rate $\lambda = 8.2137$.

Figures 9, 10, 11, 12 and 13 illustrate selected performance measures under this *MAP*.

Ignoring batch arrivals and modelling the system with the matched *MAP* leads to a severe underestimation of key loss probabilities and a corresponding overestimation of system profitability. Under the *BMAP*, the minimum primary loss probability due to insufficient servers ($\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$) is approximately 0.035 under restrictive thresholds and increases to nearly 0.049 for permissive policies; in contrast, this probability never exceeds 0.013 under the *MAP*. Impatience-related ($\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}$) and total loss probabilities ($\mathrm{P}_{\mathrm{lost}}$) are likewise substantially lower in the *MAP*-based model.

As a consequence, the optimal net revenue is overstated with a maximum value of about 6.245 attained at thresholds (10,9) under the *MAP*, compared to a maximum of about 4.57 under the *BMAP*. This discrepancy indicates that *MAP*-based models not only misestimate performance measures but also lead to overly suboptimal threshold selections. These results demonstrate that explicit modeling of batch arrivals via *BMAP* is essential for reliable performance evaluation and threshold optimization in bursty and correlated traffic environments.

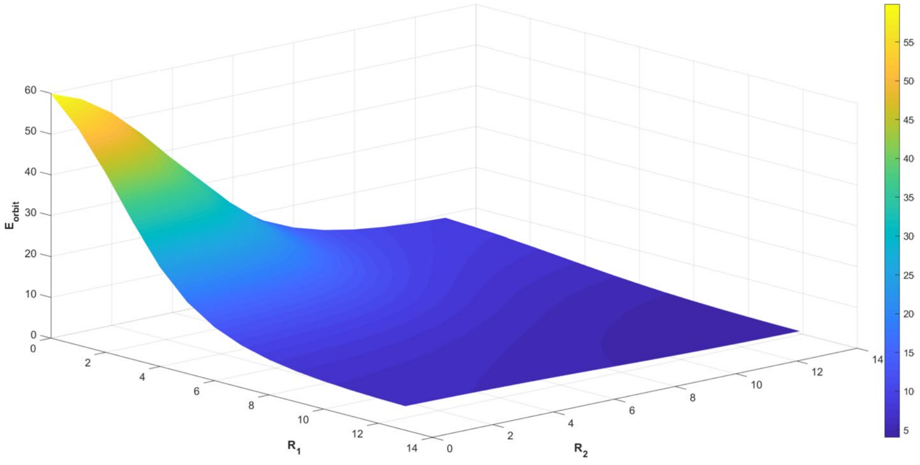


Fig. 9 Influence of R_1 and R_2 on $\mathrm{E}_{\mathrm{orbit}}$ for *MAP* arrivals

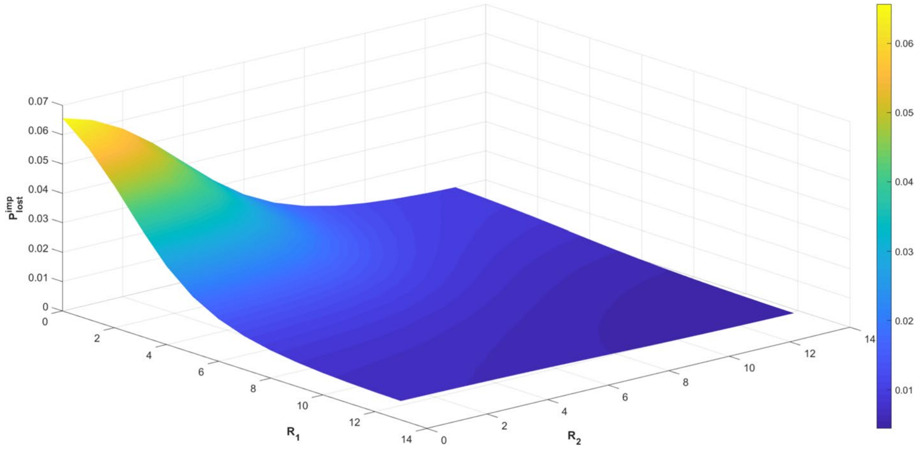


Fig. 10 Impact of R_1 and R_2 on $\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}$ for *MAP* arrivals

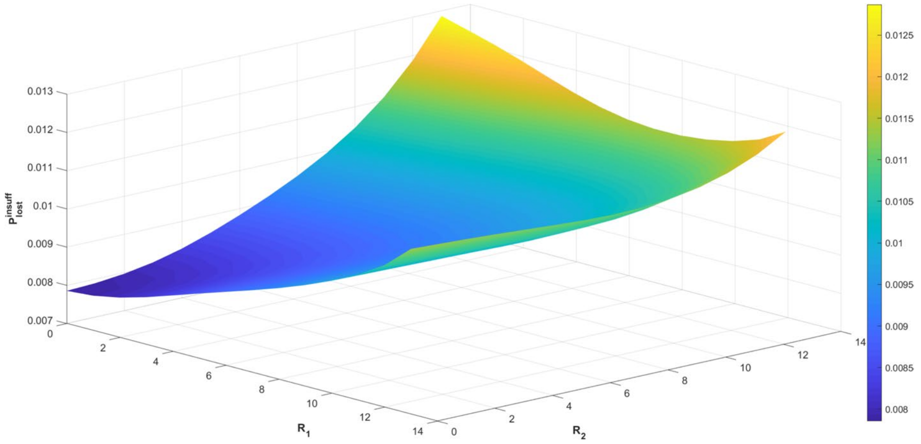


Fig. 11 Effect of R_1 and R_2 on $\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$ for *MAP* arrivals

Remark Based on the optimization results obtained in Illustration 1, the retrial thresholds are fixed at their near-optimal values $(R_1, R_2) = (9, 8)$ in Illustrations 3–6. This allows us to isolate and systematically examine the sensitivity of system performance to other model parameters without confounding effects from suboptimal threshold selection.

6.3 Illustration 3. Sensitivity to Synchronous Working Vacation Parameters

To investigate the sensitivity of system performance to the synchronous working vacation mechanism and to quantify the joint impact of vacation duration and service-rate degradation, we vary the vacation completion rate ω (with mean vacation duration $1/\omega$) and the service slowdown factor $\phi \in (0, 1)$.

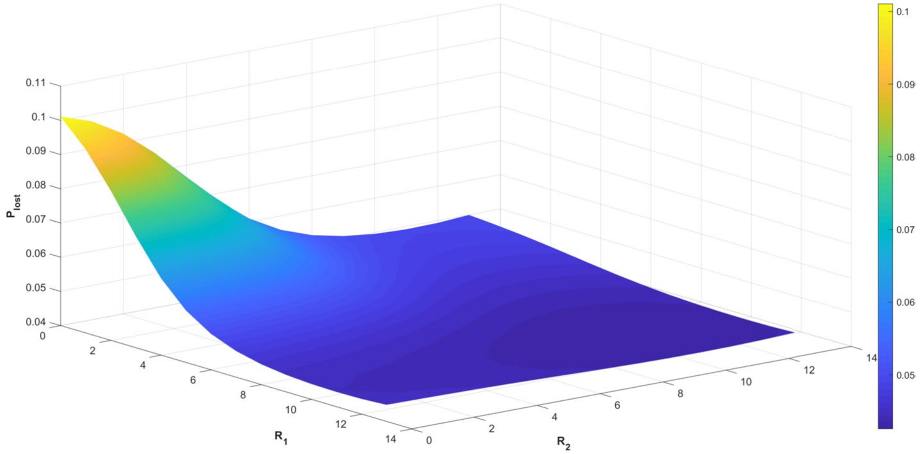


Fig. 12 Effect of R_1 and R_2 on $\mathrm{P}_{\mathrm{lost}}$ for *MAP* arrivals

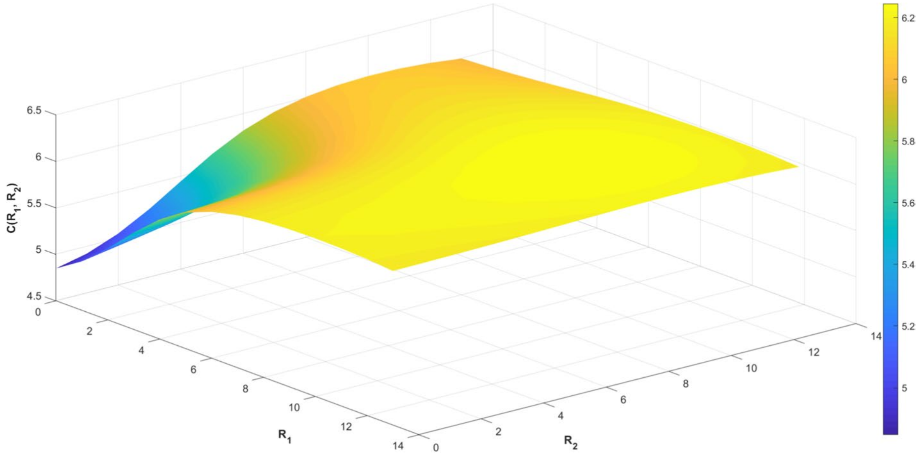


Fig. 13 Influence of R_1 and R_2 on $C(R_1, R_2)$ for *MAP* arrivals

Specifically, we consider $\phi \in \{0.1, 0.2, \ldots, 0.9\}$ and $\omega \in \{0.02, 0.05, 0.1, 0.2, 0.5, 0.75, 1, 2\}$. Smaller values of ω correspond to longer working vacation durations, while larger ω yield shorter vacations once initiated. Likewise, smaller values of ϕ represent more severe service degradation during vacations, whereas ϕ approaching unity corresponds to mild slowdown and near-normal service capacity. Figures 14 and 15 depict the joint and marginal effects of (ω, ϕ) on the mean orbit size ($\mathrm{E}_{\mathrm{orbit}}$), Figs. 16 and 17 correspond to the impatience loss probability ($\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}$), and Figs. 18 and 19 illustrate the behavior of the primary loss probability due to insufficient servers ($\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$).

The numerical results reveal pronounced interaction effects between ω and ϕ, which are further amplified by the bursty and correlated structure of the underlying *BMAP* arrivals. Both the mean orbit size, $\mathrm{E}_{\mathrm{orbit}}$, and the impatience loss probability, $\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}$, exhibit a

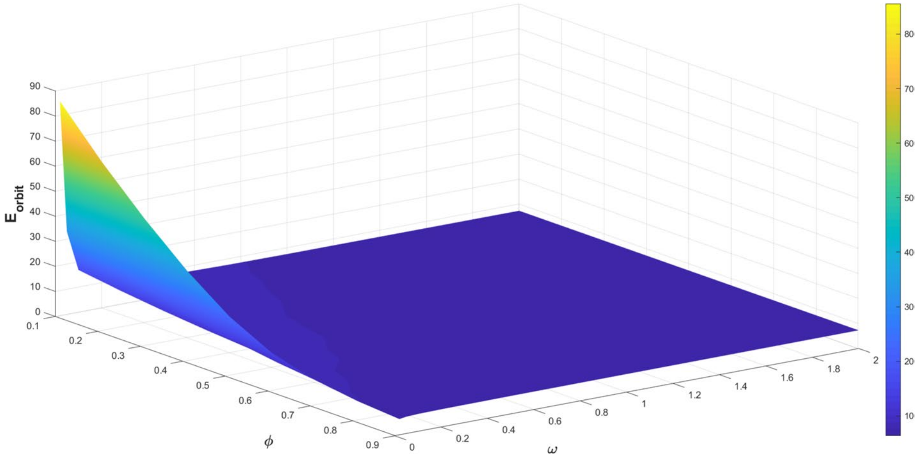
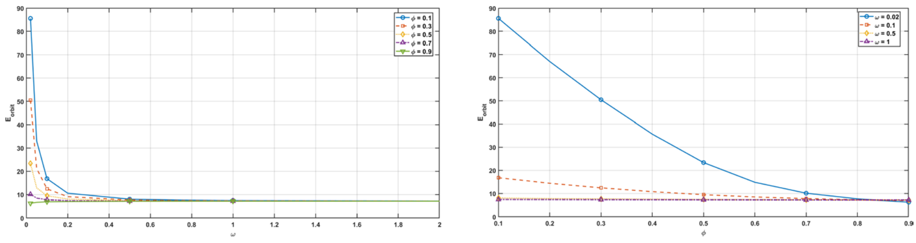


Fig. 14 Joint effect of the working vacation completion rate ω and service slowdown factor ϕ on the mean number of customers in the orbit, $\mathrm{E}_{\mathrm{orbit}}$



(a) Mean orbit size vs. ω for different ϕ

(b) Mean orbit size vs. ϕ for different ω

Fig. 15 Cross-sectional analysis of mean orbit size $\mathrm{E}_{\mathrm{orbit}}$: (a) variation with respect to vacation completion rate ω for selected service slowdown factors ϕ; (b) variation with respect to ϕ for selected values of ω

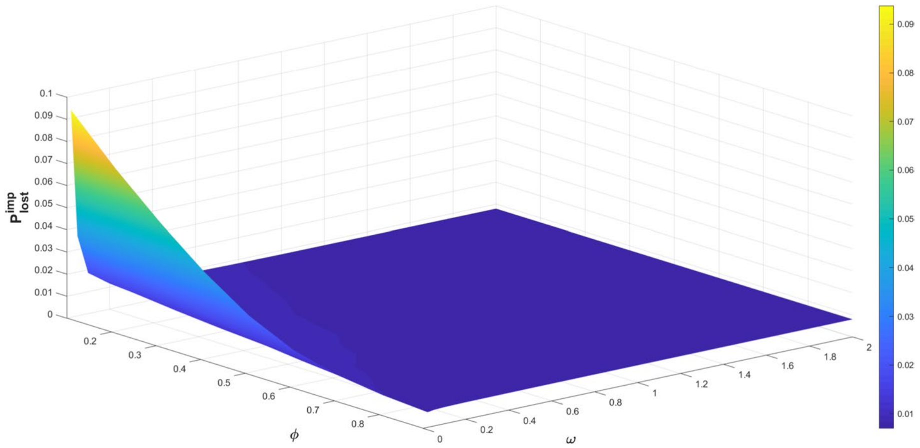
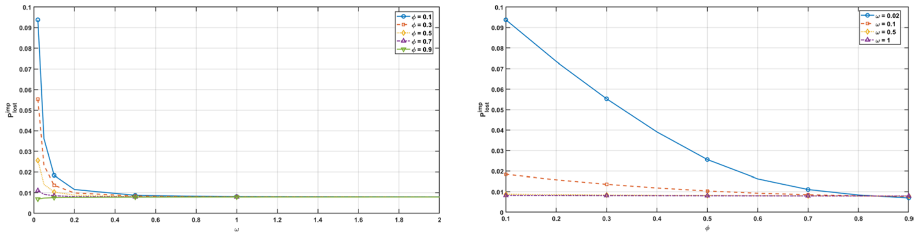


Fig. 16 Joint dependence of the impatience loss probability, $\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}$, on the working vacation parameters (ω, ϕ)



(a) $\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}$ vs. ω for different ϕ

(b) $\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}$ vs. ϕ for different ω

Fig. 17 Sensitivity analysis of impatience loss probability $\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}$: (a) variation with respect to vacation completion rate ω for selected service slowdown factors ϕ; (b) variation with respect to ϕ for selected values of ω

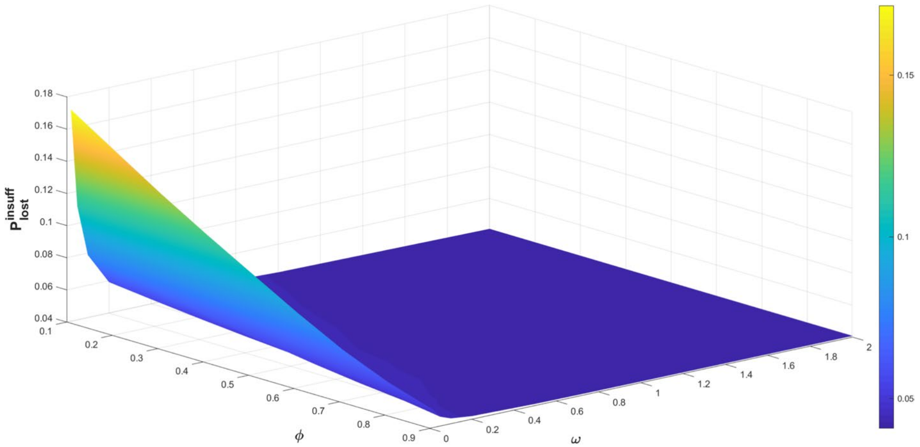
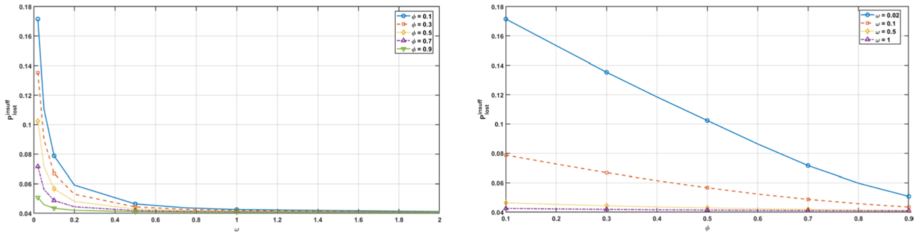


Fig. 18 Joint influence of ω and ϕ on the primary customer loss probability due to insufficient servers $\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$



(a) $\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$ vs. ω for different ϕ

(b) $\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$ vs. ϕ for different ω

Fig. 19 Sensitivity of insufficient-server loss probability $\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$ to vacation parameters: (a) variation with respect to vacation completion rate ω for selected service slowdown factors ϕ; (b) variation with respect to ϕ for selected values of ω

monotone decreasing trend as either ω or ϕ increases. In particular, under severe service slowdown ($\phi = 0.1$), long working vacations ($\omega = 0.02$) lead to substantial congestion, with $\mathrm{E}_{\mathrm{orbit}}$ exceeding 85 customers and $\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}$ approaching 0.094. In contrast, increasing ω significantly mitigates this effect by shortening vacation durations and restoring effective service capacity. For moderate to large values of $\omega \geq 0.5$ and $\phi \geq 0.5$, the orbit size stabilizes at relatively low levels (approximately 7–8 customers), indicating robust congestion control.

Conversely, the primary loss probability due to insufficient servers, $\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$, increases under longer and more severe vacations. Reduced service rates prolong server occupancy and heighten the risk of batch rejection, particularly during high-intensity arrival bursts. This effect is most pronounced when both ω and ϕ are small, but becomes negligible as ϕ approaches unity, even for moderate vacation frequencies.

Overall, these findings demonstrate that extreme vacation policies characterized by long vacation durations combined with severe service degradation are highly detrimental to system performance. In contrast, policies that maintain relatively mild service slowdown during vacations, together with sufficiently frequent vacation completions, achieve a favorable balance between orbit congestion control and primary customer protection. The results highlight the importance of jointly tuning the working vacation parameters ω and ϕ, rather than adjusting them independently, in order to sustain stable and efficient operation under bursty arrival conditions.

6.4 Illustration 4. Sensitivity to Disaster and Repair Dynamics

To assess the impact of system disruptions on the proposed retrial queueing model, we conduct a numerical sensitivity analysis with respect to the disaster occurrence rate and the repair rate.

The disaster occurrence rate ψ is varied over $\{0.005, 0.01, 0.02, 0.05, 0.1, 0.2\}$, representing environments ranging from rare to frequent catastrophic disruptions, while the repair rate η is varied over $\{0.2, 0.5, 1, 2, 5\}$, corresponding to slow to rapid system recovery. For each (ψ, η) pair, we evaluate the mean number of customers in the orbit, the impatience loss probability, and the primary customer loss probability due to balking, which together capture the combined effects of disruption-induced clearing and service unavailability.

Figures 20, 21, 22, 23, 24, 25 illustrate the joint and marginal sensitivity of system performance to the disaster occurrence rate ψ and the repair rate η. For each performance measure, a three-dimensional surface plot is presented to capture the combined effects of (ψ, η), followed by two representative cross-sections that highlight the impact of one parameter while fixing the other at selected values.

The numerical results reveal a pronounced interaction between the disaster and repair processes. For low disaster rates (e.g., $\psi = 0.005, 0.01$), the mean orbit size decreases only moderately as η increases (from about 5.50 $\rightarrow$ 3.20 and 7.79 $\rightarrow$ 3.21, respectively). This justifies that repair rate has a modest influence when disasters are rare. For moderate to high disaster rates (e.g., $\psi = 0.05, 0.1, 0.2$), the orbit size grows dramatically under slow repair (e.g., exceeding 25, 47, and even 92 customers at $\eta = 0.2$), but drops sharply as η increases. This claims that rapid repair becomes essential under frequent disruptions.

For a fixed repair rate, both the impatience loss probability and the primary customer balking probability increase substantially as the disaster rate rises, reflecting more fre-

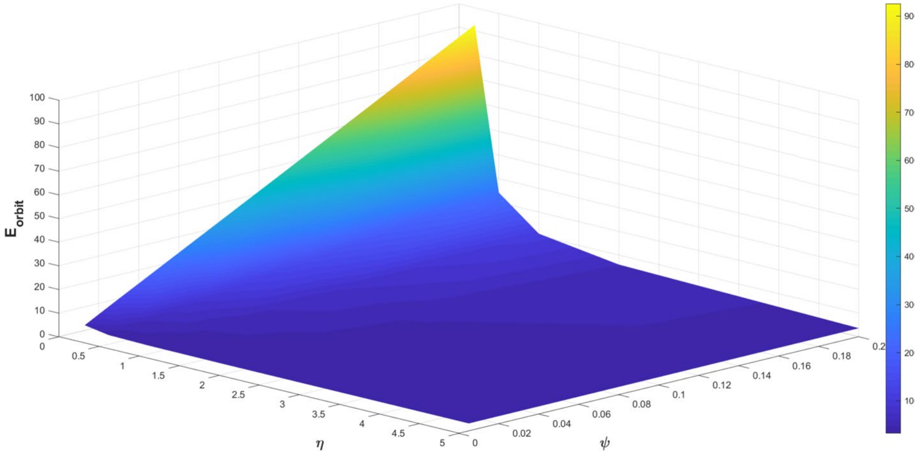
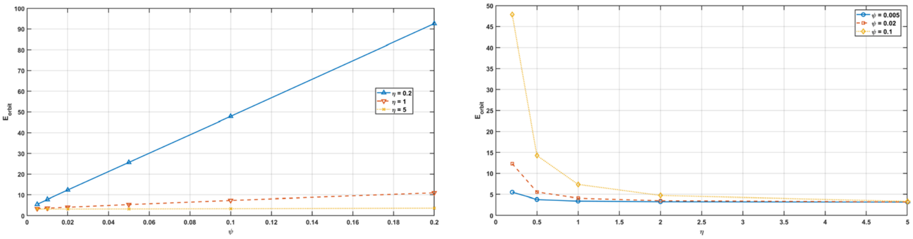


Fig. 20 Joint effect of the disaster occurrence rate ψ and the repair rate η on the mean number of customers in the orbit, $\mathrm{E}_{\mathrm{orbit}}$



(a) $\mathrm{E}_{\mathrm{orbit}}$ vs. ψ for different η

(b) $\mathrm{E}_{\mathrm{orbit}}$ vs. η for different ψ

Fig. 21 Cross-sectional analysis of mean orbit size $\mathrm{E}_{\mathrm{orbit}}$ under disaster-repair scenarios: (a) variation with respect to disaster occurrence rate ψ for selected repair rates η; (b) variation with respect to repair rate η for selected values of ψ

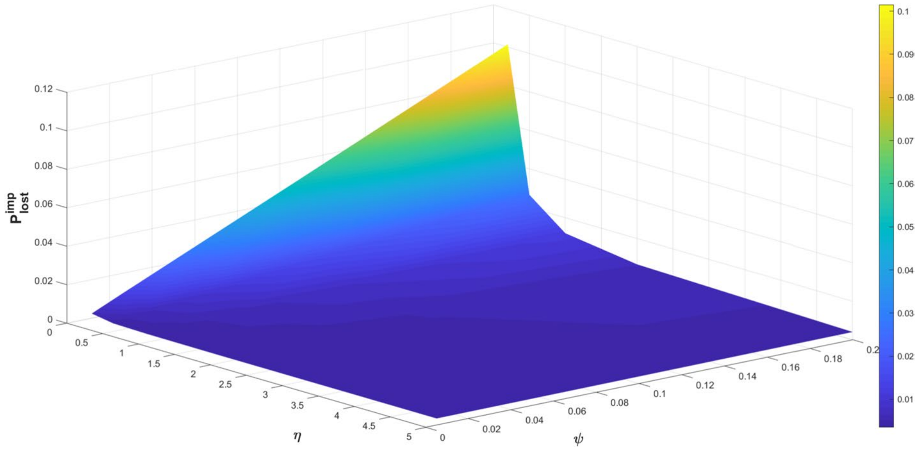
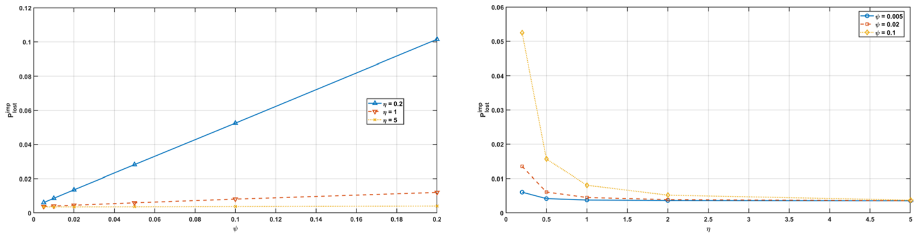


Fig. 22 Joint effect of the disaster occurrence rate ψ and the repair rate η on the impatience loss probability $\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}$



(a) P^{imp}_{lost} vs. ψ for different η

(b) P^{imp}_{lost} vs. η for different ψ

Fig. 23 Cross-sectional analysis of impatience loss probability P^{imp}_{lost} under disaster-repair scenarios: (a) variation with respect to disaster occurrence rate ψ for selected repair rates η; (b) variation with respect to repair rate η for selected values of ψ

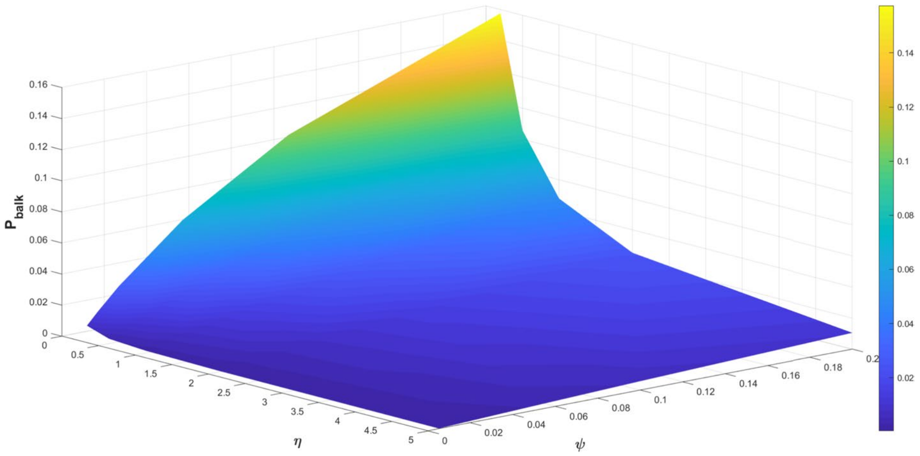
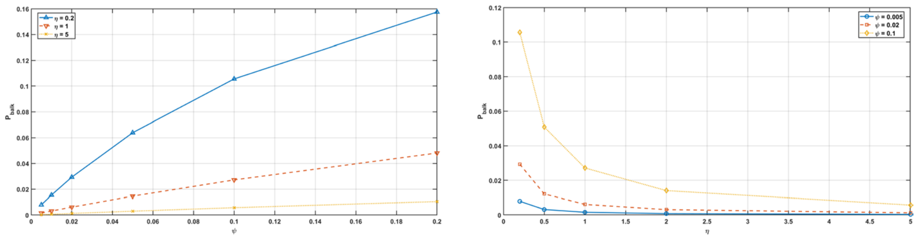


Fig. 24 Joint influence of the disaster occurrence rate ψ and the repair rate η on the primary customer loss probability due to balking, P_{balk}



(a) P_{balk} vs. ψ for different η

(b) P_{balk} vs. η for different ψ

Fig. 25 Sensitivity of balking loss probability P_{balk} to disaster-repair parameters: (a) variation with respect to disaster occurrence rate ψ for selected repair rates η; (b) variation with respect to repair rate η for selected values of ψ

quent service interruptions. This effect is particularly pronounced under slow repair, where prolonged service unavailability leads to sustained orbit buildup and higher customer abandonment.

Increasing the repair rate η significantly mitigates both forms of loss. For each disaster rate, faster repair results in sharp reductions in impatience losses by limiting waiting times in the orbit, and simultaneously lowers balking losses by restoring service availability more quickly. The sensitivity to η becomes stronger as disasters become more frequent, indicating a pronounced interaction between disruption intensity and recovery speed.

Overall, the results demonstrate that while disasters inherently degrade system performance, rapid repair mechanisms are highly effective in controlling both impatience induced and balking-induced losses, especially in environments subject to frequent disruptive events.

6.5 Illustration 5. Combined Effect of Retrial and Impatience Rates

To gain deeper insight into orbit customer behavior, we investigate the joint influence of the retrial rate θ (intensity of retry attempts from the orbit) and the impatience rate γ (individual abandonment rate per orbiting customer). We vary $\theta \in \{0.1, 0.2, 0.5, 1, 2\}$ and $\gamma \in \{0.01, 0.05, 0.1, 0.2, 0.5\}$.

Figure 26 presents the mean orbit size $\mathrm{E}_{\mathrm{orbit}}$ as a function of θ and γ.

The surface shows a pronounced decrease in orbit congestion as the retrial rate increases, with $\mathrm{E}_{\mathrm{orbit}}$ dropping from over 12 customers at low retrial intensity ($\theta = 0.1$) to below 2 at high intensity ($\theta = 2$). Higher impatience rates reduce the orbit size further at low θ, but this effect diminishes as θ grows, since rapid retrials shorten waiting times and leave little opportunity for abandonment.

Figure 27 displays the impatience loss probability $\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}$. This metric increases monotonically with γ (more customers abandon per unit time) but decreases with θ (faster retrials reduce waiting time in orbit).

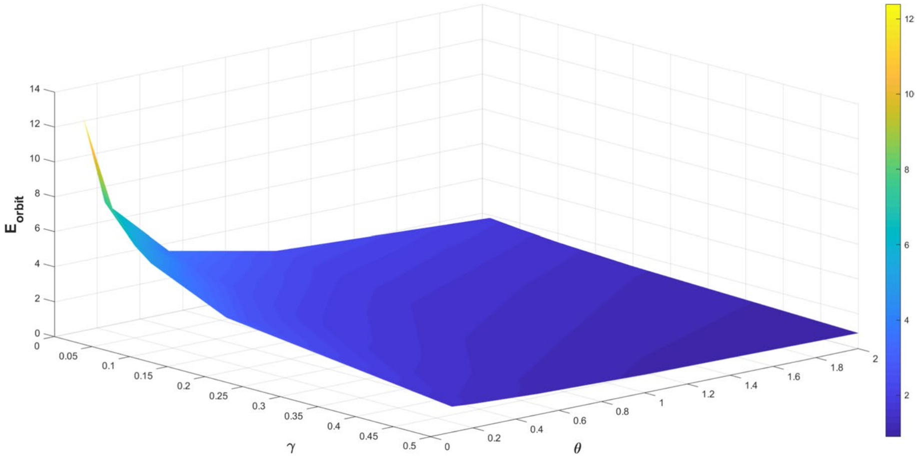


Fig. 26 Joint effect of the retrial rate θ and the impatience rate γ on the mean number of customers in the orbit, $\mathrm{E}_{\mathrm{orbit}}$

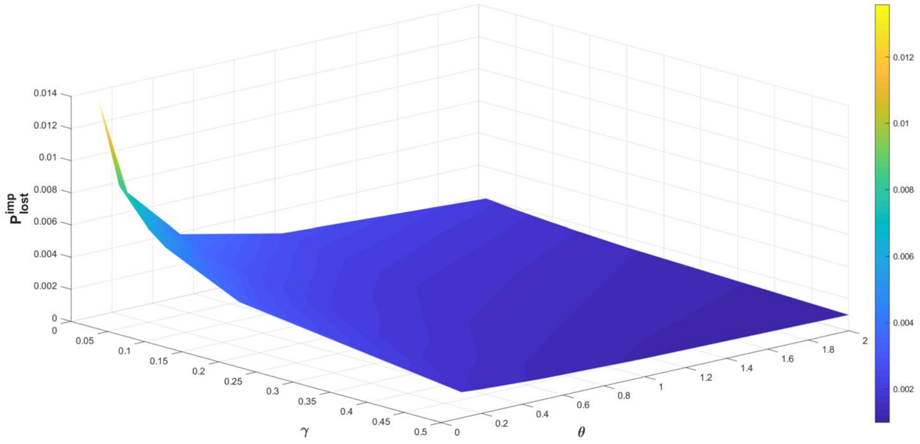


Fig. 27 Joint dependence of the impatience loss probability, $\mathrm{P}_{\mathrm{lost}}^{\mathrm{imp}}$, on the retrial rate θ and the impatience rate γ

Figure 28 shows the primary insufficient-server loss probability $\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$. It increases slightly with higher retrial rates (more aggressive orbit clearing occupies servers, mildly raising the risk for primary arrivals) and decreases with higher impatience rates.

Overall, the results highlight a manageable trade-off moderated by the bursty arrival process: higher retrial rates efficiently clear the orbit and suppress abandonment but marginally elevate primary losses due to increased server competition. In systems with correlated batch arrivals and state-dependent retrial control, tuning θ and γ allows operators to balance orbit stability against immediate service accessibility.

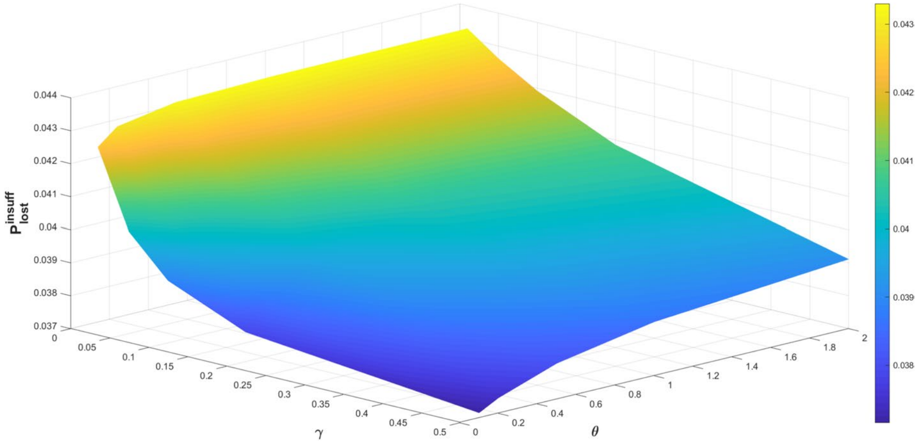


Fig. 28 Influence of the retrial rate θ and the impatience rate γ on the primary customer loss probability due to insufficient servers, $\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$

6.6 Illustration 6. Impact of Balking Behavior During Vacation and Repair Interruptions

This final illustration examines how customer balking decisions affect system performance when service capacity is limited or entirely unavailable. Specifically, we analyze:

- Vacation balking probability (b_v): The probability that an entire batch is rejected upon arrival when the servers are in vacation mode and the number of idle servers is insufficient to serve the batch.
- Repair balking probability (b_r): The probability that an entire batch is rejected when the servers are in repair mode , i.e., when service is completely unavailable.

Both mechanisms reflect real-world scenarios where customers may opt not to join a system experiencing degraded or interrupted service.

6.6.1 Effect of Vacation Balking Probability

We first examine the impact of the vacation balking probability b_v on system performance under working vacation interruptions. The vacation completion rate is varied across $\omega \in \{0.02, 0.05, 0.1, 0.2, 0.5, 0.75, 1, 2\}$. The vacation balking probability b_v is increased from 0.1 to 0.9 in steps of 0.1. Key performance measures including the mean orbit size $\mathrm{E}_{\mathrm{orbit}}$, and the probability of loss due to insufficient idle servers $\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$, are analyzed as functions of b_v and ω.

As shown in Figs. 29 and 30, the analysis reveals that both the mean orbit size $\mathrm{E}_{\mathrm{orbit}}$ and the loss probability due to insufficient servers $\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$ are significantly influenced by the joint variation of the vacation balking probability b_v and the vacation completion rate ω.

For a fixed vacation duration, increasing b_v reduces the orbit size, since more customers are rejected outright but at the cost of a higher immediate loss probability. Conversely, for a

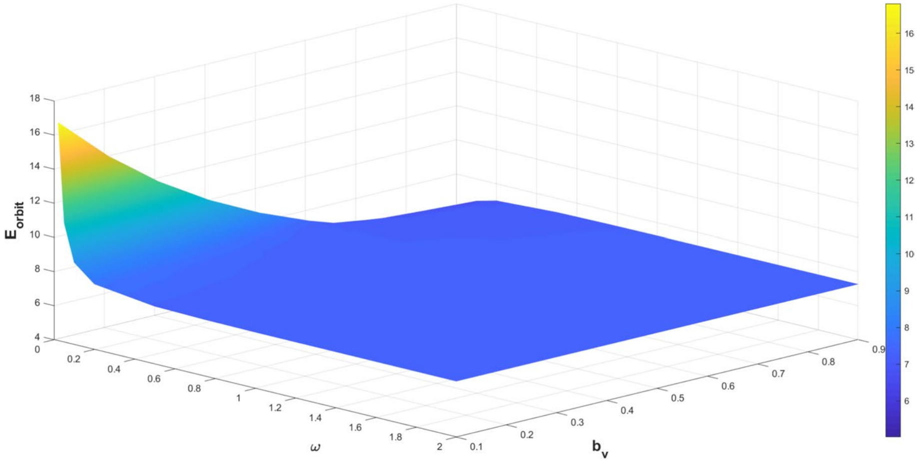


Fig. 29 Three-dimensional surface plot of the mean orbit size $\mathrm{E}_{\mathrm{orbit}}$ as a function of the vacation balking probability b_v and the vacation completion rate ω

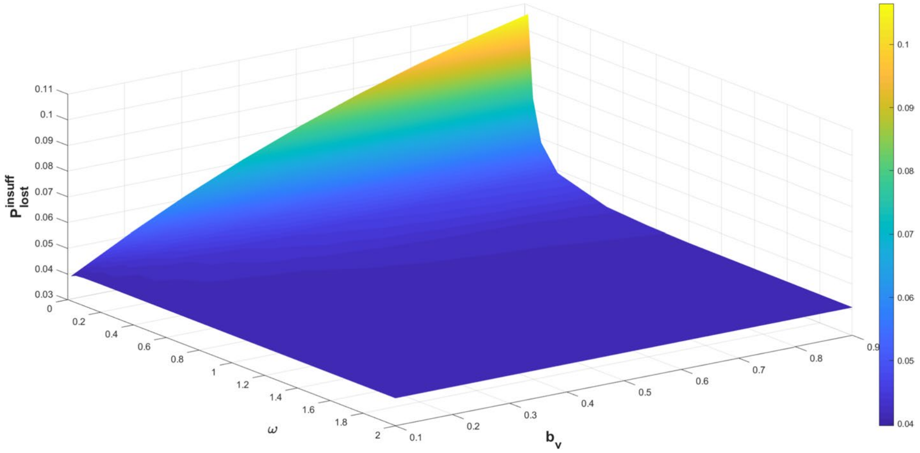


Fig. 30 Three-dimensional surface plot of the loss probability due to insufficient servers $\mathrm{P}_{\mathrm{lost}}^{\mathrm{insuff}}$ as a function of the vacation balking probability b_v and the vacation completion rate ω

fixed b_v, shorter vacations (larger ω) lower both congestion and loss by limiting the time the system operates in reduced-capacity mode.

These findings underscore the necessity of coordinating vacation duration: a moderate balking probability combined with reasonably short vacations strikes a practical balance between orbit congestion and customer loss.

6.6.2 Effect of Balking Probability During the Repair Period

We next analyze the impact of the balking probability during repair periods, denoted by b_r, on system performance. This parameter represents the likelihood that an entire arriving batch is rejected when the servers are under repair ($\tau = 2$), i.e., when service is completely unavailable.

To quantify how customer balking during repair interacts with the speed of system recovery, we vary the repair completion rate $\eta \in \{0.2, 0.5, 1, 2, 5\}$ together with the balking probability $b_r \in \{0.1, 0.2, \ldots, 0.9\}$. For each combination (η, b_r), we evaluate two key performance measures: the mean orbit size $\mathrm{E}_{\mathrm{orbit}}$ and the overall balking loss probability $\mathrm{P}_{\mathrm{balk}}$.

The joint influence of the repair-period balking probability b_r and the repair completion rate η is visualized in Figs. 31 and 32. Figure 31 illustrates that the mean orbit size $\mathrm{E}_{\mathrm{orbit}}$ forms a steep surface that declines sharply with both increasing b_r and increasing η. When repairs are slow (η small), the orbit grows substantially unless customers are highly likely to balk ($b_r \approx 1$). Conversely, for fast repairs ($\eta \geq 2$), the orbit remains small even when balking is rare, indicating that rapid recovery prevents backlog accumulation.

Figure 32 shows that the balking loss probability $\mathrm{P}_{\mathrm{balk}}$ exhibits a complementary behavior: it rises monotonically with b_r and falls as η grows. The steepest increase occurs under slow repairs ($\eta = 0.2$), where raising b_r from 0.1 to 0.9 elevates $\mathrm{P}_{\mathrm{balk}}$. In contrast, for high repair rates ($\eta = 5$), the loss surface is nearly flat, reflecting that fast repair limits the exposure of arrivals to the repair state.

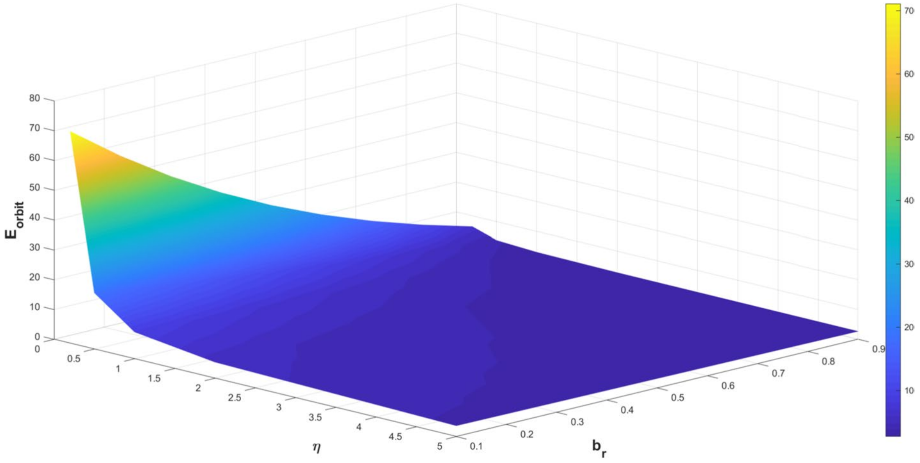


Fig. 31 Three-dimensional surface plot of the mean orbit size $\mathrm{E}_{\mathrm{orbit}}$ as a function of the balking probability during repair b_r and the repair completion rate η

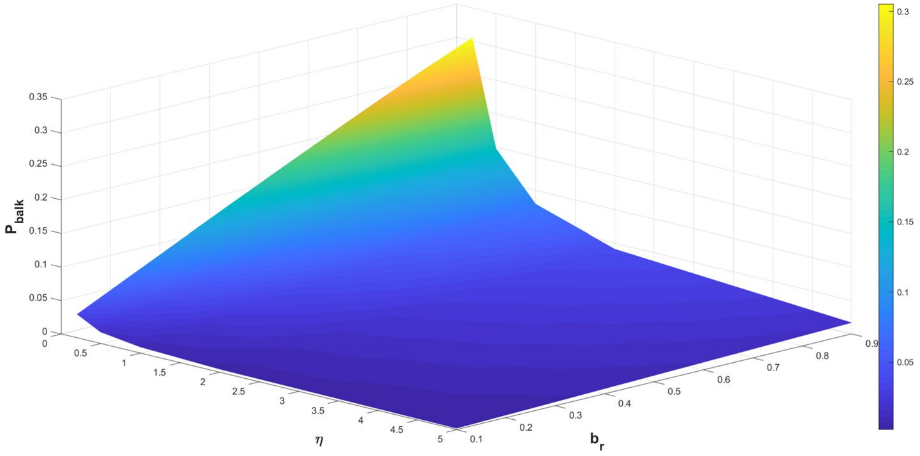


Fig. 32 Three-dimensional surface plot of the balking loss probability $\mathrm{P}_{\mathrm{balk}}$ as a function of the repair balking probability b_r and the repair completion rate η

Together, the surfaces highlight a clear design trade-off: in slowly repaired systems (η small), managers must choose between a large orbit (low b_r) and high immediate loss (high b_r). When repairs are fast ($\eta \geq 2$), this trade-off largely disappears, as both performance metrics remain at acceptable levels across the whole range of b_r. This emphasizes the value of investing in rapid recovery mechanisms, especially in systems with correlated batch arrivals and state-dependent retrial control to maintain robustness without sacrificing customer retention.

7 Conclusion

Developing a new and very broad model including many important aspects of modern sophisticated systems, this study has further advanced the idea of multi-server retrial queues. The suggested approach uses a batch Markovian arrival process (*BMAP*) to account for correlated and bursty arrivals, and phase-type (*PH*) distributed service times, disaster and synchronous working vacations. This model uses a state-dependent retrial admission control policy, where orbiting customers are given access based on thresholds linked to the underlying *BMAP* phase changed dynamically. This enables smart resource prioritization, therefore improving the efficiency and flexibility of the system. The study shows that a structured multidimensional Markov chain can capture the complex behavior of the system. Customer impatience, a reasonable assumption that guarantees the ergodicity of the underlying stochastic process and so guarantees the stability of the system across all parameter values. We also defined the fundamental performance indicators of the system. As shown, our numerical analyzes show that selecting optimum thresholds can help to improve operational efficiency. Furthermore, the studies highlight the need to carefully consider batch arrivals to avoid suboptimal outcomes. We also illustrated the effect of various system parameters on key performance measures in our model. To enhance the model's utility, adding batch processing capabilities could be useful. Incorporating heterogeneous servers with different service speeds and potential failure during the service process would better simulate real-world distributed systems.

Author Contributions All authors contributed equally to the conceptualization and discussion of the model, performed the computations and analyzed the results. All authors reviewed and approved the final manuscript.

Funding No funding was obtained for this study.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Ethical Approval Not applicable.

Competing Interests The authors declare no competing interests.

References

Ammar SI, Rajadurai P (2019) Performance analysis of preemptive priority retrial queueing system with disaster under working breakdown services. Symmetry 11:419. https://doi.org/10.3390/sym11030419

Artalejo JR, Gómez-Corral A (2008) Retrial queueing systems: A computational approach, Springer, Berlin-Heidelberg. https://doi.org/10.1007/978-3-540-78725-9

Ayyappan G, Gowthami R (2021) Analysis of $MAP/PH/1$ retrial queue with constant retrial rate, working vacations, abandonment, flush out, search of customers, breakdown and repair. Intl J Oper Res 42:310. https://doi.org/10.1504/ijor.2021.119409

Ayyappan G, Thilagavathy K (2024) Analysis of MAP/PH$_1$, PH$_2$/2 queueing system with two types of heterogeneous servers, standby server, instantaneous feedback, working vacation, multiple vacations, breakdown, phase type repairs and impatient behaviour of customers. Intl J Serv Oper Manag 48:353–394. https://doi.org/10.1504/ijsom.2024.139241

Breuer L, Dudin A, Klimenok V (2002) A retrial $BMAP/PH/N$ system. Queueing Syst 40:433–457. https://doi.org/10.1023/a:1015041602946

Dimitriou I (2013) A mixed priority retrial queue with negative arrivals, unreliable server and multiple vacations. Appl Math ModelL 37:1295–1309. https://doi.org/10.1016/j.apm.2012.04.011

Dudin AN, Dudin SA, Klimenok VI, Dudina OS (2024) Stability of queueing systems with impatience, balking and non-persistence of customers. Math 12:2214. https://doi.org/10.3390/math12142214

Dudin SA, Dudina OS, Imomov AA, Kopats DY (2025) Analysis of $BMAP/PH/N$-type queueing system with flexible retrials admission control. Mathe 13:1434. https://doi.org/10.3390/math13091434

Dudin S, Dudin A, Kostyukova O, Dudina O (2020) Effective algorithm for computation of the stationary distribution of multi-dimensional level-dependent Markov chains with upper block-Hessenberg structure of the generator. J Comput Appl Math 366:112425. https://doi.org/10.1016/j.cam.2019.112425

Dudin A, Krishnamoorthy A, Dudin S, Dudina O (2024) Queueing system with control by admission of retrial requests depending on the number of busy servers and state of the underlying process of Markov arrival process of primary requests. Ann Oper Res 335:135–150. https://doi.org/10.1007/s10479-023-05728-1

Falin G (1990) A survey of retrial queues. Queueing Syst 7:127–167. https://doi.org/10.1007/bf01158472

Falin GI, Templeton JGC (1997) Retrial Queues. Chapman and Hall, London

Gao S, Zhang D, Dong H, Wang X (2021) Equilibrium balking strategies in the repairable $M/M/1$ G-retrial queue with complete removals. Prob Eng Inf Sci 35:138–157. https://doi.org/10.1017/S026996481900010X

GnanaSekar MMN, Kandaiyan I (2022) Analysis of an $M/G/1$ retrial queue with delayed repair and feedback under working vacation policy with impatient customers. Symmetry 14:2024. https://doi.org/10.3390/sym14102024

Gupta P, Kumar N (2021) Performance analysis of retrial queueing model with working vacation, interruption, waiting server, breakdown and repair. J Sci Res 13:833–844. https://doi.org/10.3329/jsr.v13i3.52546

He QM, Alfa AS (2018) Space reduction for a class of multidimensional Markov chains: A summary and some applications. INFORMS J Comput 30:1–10. https://doi.org/10.1287/ijoc.2017.0759

Ke JC, Chang FM, Liu TH (2024) Bi-objective optimization of a retrial model with synchronous working vacation interruption. Quality Technol Quant Manag 22:659–682. https://doi.org/10.1080/16843703.2024.2380953

Kim CS, Dudin S, Taramin O, Baek J (2013) Queueing system $MAP|PH|N|N+R$ with impatient heterogeneous customers as a model of call center. Appl Math Modell 37:958–976. https://doi.org/10.1016/j.apm.2012.03.021

Kim CS, Klimenok V, Mushko V, Dudin A (2010) The $BMAP/PH/N$ retrial queueing system operating in Markovian random environment. Comput Operat Res 37:1228–1237. https://doi.org/10.1016/j.cor.2009.09.008

Kim CS, Park SH, Dudin A, Klimenok V, Tsarenkov G (2010) Investigation of the $BMAP/G/1 \to \cdot/PH/1/M$ tandem queue with retrials and losses. Appl Math Modell 34:2926–2940. https://doi.org/10.1016/j.apm.2010.01.003

Kim C, Klimenok VI, Orlovsky DS (2008) The $BMAP/PH/N$ retrial queue with Markovian flow of breakdowns. Eur J Oper Res 189:1057–1072. https://doi.org/10.1016/j.ejor.2007.02.053

Klimenok VI, Orlovsky DS, Dudin AN (2007) A $BMAP/PH/N$ system with impatient repeated calls. Asia Pacific J Oper Res 24:293–312. https://doi.org/10.1142/s0217595907001310

Klimenok V, Dudin A (2006) Multi-dimensional asymptotically quasi-Toeplitz Markov chains and their application in queueing theory. Queueing Syst 54:245–259. https://doi.org/10.1007/s11134-006-0300-z

Latouche G, Ramaswami V (1999) Introduction to matrix analytic methods in stochastic modeling. SIAM. https://doi.org/10.1137/1.9780898719734

Li J, Li T (2020) An $M/G/1$ G-queue with server breakdown, working vacations and Bernoulli vacation interruption. IAENG Int J Appl Math 50:421–428

Lisovskaya E, Fedorova E, Salimzyanov R, Moiseeva S (2022) Resource retrial queue with two orbits and negative customers. Math 10:321. https://doi.org/10.3390/math10030321

Liu TH, Chiou KC, Chen CM, Chang FM (2024) Multiserver retrial queue with two-way communication and synchronous working vacation. Math 12:1163. https://doi.org/10.3390/math12081163

Li T, Zhang L, Gao S, (2018) An $M/G/1$ retrial queue with balking customers and Bernoulli working vacation interruption. Quality Technol Quant Manag 16:511–530. https://doi.org/10.1080/16843703.2018.1480264

Lucantoni DM (1991) New results on the single server queue with a batch Markovian arrival process. Commun Stat Stoch Models 7:1–46. https://doi.org/10.1080/15326349108807174

Neuts MF (1979) A versatile Markovian point process. J Appl Probab 16:764–779. https://doi.org/10.2307/3213143

Neuts MF (2021) Structured stochastic matrices of $M/G/1$ type and their applications, CRC Press: Boca Raton. FL, USA

Subramanian MG, Govindan A, Sekar G (2011) Study of multi server retrial queueing system under vacation policies by direct truncation method. In: Proceedings of the international conference on advances in computing, communication and control, pp 169–176. https://doi.org/10.1145/2021216.2021241

Yang T, Templeton JGC (1989) A survey on retrial queues. Queueing Syst 4(94). https://doi.org/10.1007/BF01150861

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.