

Article

Multi-Objective Optimization of a Multi-Server Retrial Machine Repair System with Orbital Search and Synchronous Vacation

Lee-Wen Chiu ¹, Ming-Chin Chen ¹, Tzu-Hsin Liu ^{2,*} and Fu-Min Chang ^{2,*}

¹ Ph.D. Program of Business Administration in Industrial Development, Department of Business Administration, Chaoyang University of Technology, 168, Jifeng East Road, Wufeng District, Taichung City 41349, Taiwan; s11437905@o365.cyut.edu.tw (L.-W.C.); s11137905@o365.cyut.edu.tw (M.-C.C.)

² Department of Finance, Chaoyang University of Technology, 168, Jifeng East Road, Wufeng District, Taichung City 41349, Taiwan

* Correspondence: t2011119@cyut.edu.tw (T.-H.L.); fmchang@cyut.edu.tw (F.-M.C.)

Abstract

This paper investigates a multi-server retrial machine repair system that incorporates orbital search and a synchronous vacation mechanism. The system features standby units and examines two potential vacation scenarios for servers, reflecting real-world situations, such as technical staff in teaching hospitals taking periodic administrative or training vacations. We formulate a mathematical model using birth-and-death processes to establish the governing equation and propose a recursive matrix method to systematically derive the steady-state probabilities. Performance measures, including system availability, the expected number of failed units in the orbit, and expected waiting times, are derived. To address the conflicting objectives of minimizing total operating costs, minimizing expected waiting times, and maximizing system availability, we construct a tri-objective optimization problem. By implementing a multi-objective genetic algorithm, we identify a set of Pareto-optimal frontiers and reveal the explicit financial and operational trade-offs among these competing criteria. Numerical experiments and sensitivity analyses demonstrate that enhancing the automated retrial rate and managing the emergency repair rate are most critical to minimizing system downtime. Furthermore, the joint optimization of server capacities and vacation schedules effectively eliminates operational redundancy, showing that near-perfect equipment availability (up to 0.999) can be achieved with only marginal increases in cost. This research provides administrators with a robust decision-making framework to optimize technical resource management while ensuring near-perfect equipment availability in real-world environments.

Keywords: retrial machine repair system; multi-objective genetic algorithm; synchronous vacation; orbital search

Received: 29 May 2026

Revised: 18 June 2026

Accepted: 29 June 2026

Published: 2 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Machine repair systems serve as practical mathematical models applicable to a wide range of fields, including telecommunications, manufacturing, and cloud computing, where multiple servers provide services to customers or products. In intensive care units (ICUs), the availability of critical medical equipment, such as ventilators, directly affects

patient survival and the quality of healthcare delivery. However, equipment failures are inevitable due to their stochastic nature, making it a significant challenge for hospitals to maintain an adequate supply of operational equipment under limited resources. Furthermore, in teaching hospitals, biomedical engineers are responsible not only for equipment maintenance but also for administrative duties, teaching activities, and professional training. As a result, maintenance personnel may occasionally be unavailable because of training programs, administrative assignments, or vacations, leading to a temporary reduction in maintenance capacity. Therefore, one of the primary motivations of this study is to investigate how such vacation mechanisms influence the availability of critical medical equipment and, consequently, ICU operational safety.

In traditional retrial queueing systems, failed units enter a retrial orbit and repeatedly attempt to obtain service according to a stochastic retrial process. In healthcare environments, however, this passive “wait-and-retrial” mechanism may introduce undesirable delays, during which life-supporting devices remain unavailable despite the presence of idle maintenance personnel. To address this limitation, the concept of orbital search is incorporated into the proposed model. This mechanism enables technicians to proactively identify and retrieve failed units from the orbit for immediate repair, thereby transforming maintenance operations from a reactive approach into a proactive one. Such an approach is expected to enhance equipment availability, maximize the operational fleet size of critical medical devices, and ultimately improve the reliability and resilience of ICU services.

1.1. Literature Review

Extensive research has been conducted on retrial behavior in machine maintenance systems and queueing systems. In the realm of queueing theory, contemporary literature has significantly advanced the operational modeling of multi-server systems with repeated attempts. For instance, D’Apice et al. [1] explored a multi-server retrial queue characterized by fluctuating finite request sources. Similarly, Dudin et al. [2] investigated a multi-server retrial queue featuring a batch Markov arrival process and a phase-type service time distribution. Beyond traditional queueing, scholars have extensively examined how hybrid standby components, repeated retries, and retrial dynamics influence key reliability metrics, such as machine availability and overall system reliability. Notably, Kuo et al. [3] examined the impact of retrial behavior on system reliability measures in machine repair systems with mixed standby components, comparing the cost-effectiveness of systems with and without retrials. Additional significant contributions to complex machine repair systems can be found in [4–11].

Recent studies, such as the one by Wu et al. [12], investigated limited-capacity machine repair problems with standbys, two failure modes, and working vacation policies. Shao et al. [13] developed a reliability model for a k -out-of- n : G mixed standby retrial system featuring failure dependency and a J-vacation policy. Chen et al. [14] discussed reliability and sensitivity analyses of a retrial machine repair system with warm standby units and a single repair server capable of taking multiple vacations when no failed units are present. Mathavavisakan et al. [15] presented a comprehensive model for the steady-state behavior of an M/G/1 retrial queue with fault correction and hybrid vacation policies, addressing real-world challenges, such as system breakdowns, repair delays, and cost optimization. Shao et al. [16] developed a reliability model for a (k_1, k_2) -out-of- n : G mixed standby retrial system with redundant dependencies and working vacations. Wu & Wang [17] examined a retrial machine repair system featuring multiple unreliable machines with two failure modes, providing an analytical framework and strategic insights for optimal decision-making.

In some scenarios, idle servers can inform customers in orbit about their status, allowing the server to potentially retrieve a customer directly from the orbit with some probability once service is completed, provided no other customers are waiting in the queue. This behavior, known as orbit search, was first proposed by Neuts & Ramalhoto [18] in classical queues. Ahammed et al. [19] studied a single-server retrial queuing system with two types of service under modified vacation policies, incorporating orbit search and breakdown. Li & Li [20] considered a retrial queue with working vacations, orbit search, and balking, deriving stability conditions and steady-state distributions using matrix analytic methods. Nila & Sumitha [21] analyzed a batch arrival retrial queue with preemptive resume priority, multi-optional second phases, feedback, and orbital search. Gao & Wang [22] examined a preemptive priority retrial queue with orbital search and multiple exhaustive vacations, discussing necessary and sufficient conditions for system stability and performance measures using embedded Markov chain techniques and supplementary variable methods. Kumar et al. [23] investigated the steady-state behavior of a finite orbit capacity multiserver call center retrial queue with a Bernoulli vacation schedule, where servers accept incoming calls and make outgoing calls after exponentially distributed idle times.

By establishing a tactical connection among the state-of-the-art studies summarized in Tables 1 and 2, a distinct research gap explicitly emerges. On one hand, the contemporary literature on retrial queueing systems (Table 1) excels at capturing complex behavior dynamics, such as strategic admission controls, preemptive priorities, or two-way communications; yet, it frequently neglects the crucial operational interplay of physical warm standby buffers and specialized machine repair configurations. On the other hand, existing studies on machine repair systems (Table 2) thoroughly address standby redundancies and server vacation variations, including working vacations, Bernoulli schedules, or flexible threshold adjustments. Nevertheless, they typically assume that failed technical assets wait in organized queues or simple failure structures, often overlooking the practical complications of automated retrial processes combined with orbital searching mechanisms.

Our paper uniquely bridges these two historically separate sub-fields. By integrating a warm standby machine repair system directly with an orbital retrial framework and overlaying a highly specific, multi-type synchronous vacation mechanism (accurately capturing the dual administrative/training and on-call rest duties of hospital biomedical engineers), this study fills a critical academic void. Furthermore, while the majority of existing optimization works listed in both tables rely on single-objective tools or classic metaheuristics (e.g., [5,13,15,24–26]), we deploy a specialized tri-objective non-dominated multi-objective genetic algorithm (MOGA) to simultaneously evaluate the trade-offs among operational costs, customer waiting times, and mechanical availability. Tailored precisely for hospital ICU ventilator configurations under strict safety thresholds, this unique convergence fully guarantees the significant novelty and clinical contribution of our work.

Table 1. Comparative summary of existing literature on retrial queueing systems.

Reference	Considerations/Key Features	Optimization Tool	Practical Use Cases/Domain
D’Apice et al. [1]	Multi-server retrial queue, fluctuating finite sources	Matrix-analytic method	Wireless communication cells
Dudin et al. [2]	BMAP/PH/N-type system, flexible control of retrials	Direct numerical calculation	Smart manufacturing systems
Chang et al. [24]	Retrial, unreliable servers, feedback, impatience	Quasi-Newton method, pattern search method,	Network service system

		Nelder–Mead simplex direct search method	
Ammar & Rajadurai [27]	Preemptive priority, disaster, and working breakdown services	Supplementary variable technique	Computer processing systems
Saravanan et al. [28]	Optional service, unreliable server, balking, feedback	Recursive technique	Electric vehicle charging station
Shanmugam & Saravananarajan [29]	Retrial, starting failure, two-phase service, working vacation	Supplementary variable technique	Customer support ticketing system
Sundararaman et al. [30]	Two types of waiting queues, working vacation	Supplementary variable technique	Healthcare service systems
Liu et al. [31]	Two-way communication, disaster, synchronous working vacation	Multi-objective genetic algorithm	Health helpline center
Xu et al. [25]	Reserved idle times, setup time, server breakdowns	Non-dominated ranking genetic algorithm II	Cloud computing data center
Kumar et al. [26]	Retrial, discouragement, unreliable server, multiple working vacations	Particle swarm optimization, GA	Healthcare systems
This paper	Multi-server retrial, orbital search, multi-type synchronous vacations, warm standbys	MOGA	Hospital ICU ventilator maintenance

Table 2. Comparative summary of the existing literature on machine repair systems.

Reference	Considerations/ Key Features	Optimization Tool	Practical Use Cases/ Domain
Yang & Tsao [4]	Repairable system, standbys, retrial, working vacation	Matrix-analytic method	manufacturing of solar panels
Jain & Sanga [5]	Machine repair, general retrieval, admission control, fuzzy environment	Genetic algorithm (GA)	
Liu et al. [6]	Mixed standbys, Bernoulli vacation, retrial	Laplace transform	Intelligent automatic system
Gao & Wang [7]	Mixed standbys, preventive maintenance, active breakdowns	parabolic method	Ventilation safety equipment system
Sanga & Charan [9]	Retrial, admission control F-policy, feedback	Harmony search method, quasi-Newton method	Automobile repair shop
Kumar et al. [11]	Threshold recovery policy, unreliable servers, phase repairs	Greedy selection method, golden section method	Printed circuit boards
Shao et al. [13]	Mixed standbys, retrial, failure dependency, J-vacation	GA	Wind–solar storage intelligent power generation systems
Mathavavisakan et al. [15]	Hybrid vacation policy, fault correction, retrial	Gray wolf optimizer, bat algorithm, whale optimization, PSO, GA, artificial bee colony	Production-on-demand system
Shao et al. [16]	Redundant dependency, mixed standbys, working vacation	PSO	Hyper-converged infrastructure

Wu & Wang [17]	Warm standbys, two failure modes	Non-dominated Sorting GA	Compressed air system
This paper	Machine repair, orbital search, automated retrials, administrative and on-call vacations	Tri-objective MOGA	Mission-critical hospital ICU staffing

1.2. Contribution

Previous studies on retrial machine repair systems often treat maintenance as a simple, linear process, rarely integrating orbital search (automated retrials combined with proactive searches) with synchronous vacations. Additionally, most existing research focuses solely on minimizing costs or maximizing availability, neglecting a tri-objective framework that simultaneously addresses cost, waiting time, and system risk in practical contexts.

The key contributions of this article are as follows:

1. A novel multi-server retrial queuing model that incorporates orbital search, warm standbys, and synchronous vacations is developed;
2. We propose a recursive matrix method that simplifies the derivation of steady-state probabilities for complex balance equations, providing a template for future researchers;
3. Our findings indicate that over-maintenance can lead to frequent system downtimes, causing temporary queuing congestion, emphasizing the importance of system stability over speed;
4. The research highlights that during administrative and training periods, the “emergency response capability” of technicians is a core factor determining system resilience, offering a clear scientific indicator for the training of medical professionals;
5. This study reveals that excessive repair speed can paradoxically increase system instability, serving as a critical warning for hospital managers focusing solely on “speed” key performance measures;
6. By providing the Pareto optimal frontier through the multi-objective genetic algorithm, administrators are presented with a “menu” of optimal strategies tailored to their specific budget and safety targets.

This paper is structured as follows: Section 2 describes a retrial machine repair system with orbital search and synchronous vacation mechanisms, demonstrating practical applications. Section 3 frames the system’s governing equations using appropriate birth-death rates. Section 4 proposes a method for systematically deriving steady-state probabilities using the recursive matrix method. Section 5 derives key performance measures in terms of steady-state probabilities and presents numerous numerical results, investigating the impact of various system parameters on performance measures and expected automated retrial trajectory waiting times. Section 6 constructs the expected cost function and proposes a tri-objective optimization problem, considering machine availability and expected waiting times, with partial non-dominated solutions obtained through a multi-objective genetic algorithm. The Pareto optimal boundary is listed. Finally, Section 7 summarizes the paper.

2. Model Description

2.1. Assumptions

Consider a repairable retrial system consisting of a total of $M+S$ machines, where M represents the primary machines, and S denotes warm standby machines. It is important to note that at least m functioning machines are required to maintain the system in normal

operation mode. The service facility is staffed with c servers. The following assumptions and notations are used to formulate this machine repair system:

- **Initial State:** At time $t = 0$, all primary machines are operating while all standby machines are in a warm standby state. The system is active, and all repairmen are idle.
- **Failure Rates:** The primary machines fail independently according to a Poisson distribution with rate λ , while warm standby machines fail at rate α . When a primary machine fails, it is immediately replaced by an available standby machine, which then becomes the active machine. The transition time for the standby machine to become a primary machine is negligible.
- **Repair Process:** The service facility with c servers repairs the failed machines. Each server operates at a repair rate μ , and the repaired component is indistinguishable from a new one. Repair times are exponentially distributed.
- **Automated Retrial Mechanism:** If a machine fails and at least one repairman is free, the failed machine is repaired immediately. If all repairmen are busy or on vacation, the newly failed machine enters the retrial orbit and attempts to access service after a random time until repair begins. The automated retrial discipline for the failed components in the orbit is FIFO, meaning only the failed component at the front of the orbit can access the repairman. Automated retrial times are governed by an exponential distribution with rate γ .
- **Vacation Mechanism:** When the system becomes empty, all repairmen take a type I vacation simultaneously. At the end of a type I vacation, if there are no failed components in the retrial orbit, all repairmen take a shorter type II vacation simultaneously. Conversely, if failed components are present, all repairmen remain idle to address these components. If no failed components are found after completing a type II vacation, the repairmen continue their type II vacation. Vacation times for both types of vacation follow the exponential distribution with rates θ_1 and θ_2 , respectively.
- **Repair Rate during Vacation:** During the type I vacation period, repairmen can repair at a lower rate μ_v . During the type II vacation period, repairmen are inactive.
- **Idle Time and Searching:** After completing a repair, each repairman remains idle for an exponentially distributed period with rate δ . During this period, any newly arriving failed machines (whether new or retrials) are repaired immediately. After the idle time, each repairman searches for failed machines in the orbit. The searching time follows an exponential distribution with rate η .
- **Independence of Random Variables:** All random variables are independent of each other.

2.2. Practical Application

To demonstrate the structural adaptability and empirical relevance of the proposed multi-server retrial machine repair framework with synchronous multi-type vacations, this section delineates two concrete, real-world engineering scenarios. Both scenarios are thoroughly grounded in recent engineering and operational literature, and their systemic workflows are visually mapped into our abstract queueing abstractions in Figure 1.

Example 1: ICU Ventilator Fleet Management

In a high-acuity ICU or a multi-theater operating suite, the availability of life-support systems—specifically ventilators or anesthesia workstations—is critical for patient survival. A large teaching hospital manages a fleet of M primary ventilators actively supporting patients in the ICU, with a failure rate λ . To accommodate sudden admissions, the hospital maintains S warm standby ventilators that are plugged in and pre-calibrated, failing at a rate α . To uphold the ICU's "Level 1" status, at least m ventilators must be operational; otherwise, the unit must divert new patients to other hospitals.

The hospital employs a biomedical engineering (BME) team of c technicians. If multiple ventilators fail simultaneously and all c technicians are occupied in the “Clean Room” repairing units, a newly failed ventilator cannot be serviced immediately and enters a “Digital Service Queue” (the retrial orbit). The system checks for an available technician every $1/\gamma$ minutes. Following FIFO discipline, the ventilator that failed first is serviced first.

To prevent burnout and optimize labor in a 24/7 teaching hospital, the BME crew adheres to a strict “Vacation” (Duty Shift) protocol:

Type I vacation (Administrative/Training): When all ventilators are functional, the c technicians move to a “Type I” state (rate θ_1) to conduct training workshops or documentation. If a ventilator fails during this time, they provide emergency repairs at a slower rate μ_v , as they must pause teaching and sanitize.

Type II Vacation (On-call rest): After their training, if the orbit is still empty, they enter a “Type II” rest period. They remain inactive (rate θ_2) until a new failure occurs, prompting their return to the repair floor.

The model introduces a “Smart Handoff” phase:

Idle Period (δ): After a technician finishes repairing a ventilator (returning it to “new” condition at rate μ), they stay “Ready-on-Floor” for a short time. If any other machine in the ICU fails during this window, they bypass the queue and fix it immediately.

Searching Time (η): If the floor remains quiet, the technician spends time (at rate η) actively searching the “retrial orbit” database to identify any units from other wards waiting for maintenance, ensuring no machine is left unattended in the queue.

Example 2: High-Availability Server Clusters

Consider a data center hosting a critical banking application. To ensure continuous operation, the IT architecture mirrors the parameters outlined above:

- The cluster comprises M primary servers actively processing transactions and S warm standby servers that are powered on but in a low-energy state, failing at a lower rate α compared to the active rate λ . The system requires at least m servers to be functional to handle peak traffic without crashing. A maintenance crew of c specialized technicians is available.
- Technicians can typically fix a server hardware issue at a rate μ . If all technicians are busy when a new server fails, that server enters a software queue (the retrial orbit). The automated system attempts to reconnect the failed unit to a technician every few minutes (the automated retrial rate γ). Due to FIFO discipline, the server that crashed first gets priority for repair.
- When all servers are operational, technicians go on type I vacation (e.g., performing light administrative paperwork; rate θ_1). During this time, if a server fails, they can still repair it, but their repair rate drops to μ_v due to distractions.
- After completing their paperwork, if there are still no broken servers, they enter a type II vacation (e.g., an off-site training break; rate θ_2), during which they are completely unreachable.
- After a technician finishes a repair, they do not immediately move to the next task. They remain in an “Idle standby” phase (rate δ) to document the fix, during which they are immediately available if a new emergency arises. Once this period ends, they spend designated “Search time” (rate η) scanning the retrial orbit for any older failed servers still awaiting assistance.

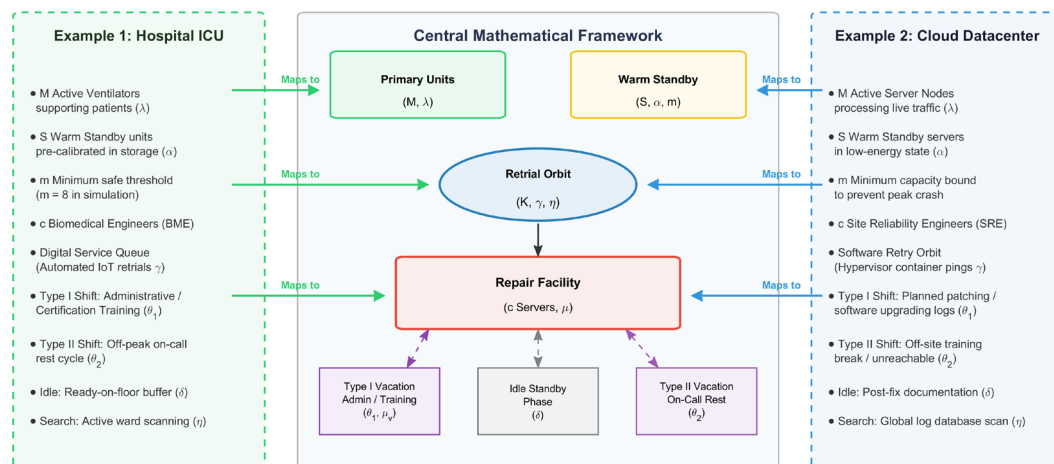


Figure 1. Comprehensive architectural mapping of practical application scenarios (ICU ventilator fleets and high-availability server clusters) into the proposed multi-server retrial machine repair framework.

3. Governing Equations

At time t , the state of the system can be described by $\{N(t), I(t), J(t), V(t); t \geq 0\}$, where

- $N(t)$: number of failed machines in the orbit at time t ;
- $I(t)$: number of servers currently repairing failed machines at time t ;
- $J(t)$: number of servers searching for failed machines at time t ;
- $V(t)$: state of the servers (0: non-vacation, 1: type I vacation, 2: type II vacation) at time t .

The stochastic process $\{N(t), I(t), J(t), V(t); t \geq 0\}$ is modeled as an irreducible regular continuous-time Markov chain with state space defined as

$$\Omega = \{(n, i, j, 0); 0 \leq n \leq K - c, 0 \leq i + j \leq c\} \cup \{(n, i, 0, 1); 0 \leq n \leq K - c, 0 \leq i \leq c\} \cup \{(n, 0, 0, 2); 0 \leq n \leq K - c\}, \text{ where } K = M + S - m + 1.$$

Define the limiting probabilities as $P_{n,i,j,v}$ as $t \rightarrow \infty$. Before going further, the following notations are used in our model.

- M : number of primary machines;
- S : number of warm standby machines;
- m : minimum number of operating machines required;
- c : number of repairmen in the repair facility;
- λ : failure rate of primary machines;
- α : failure rate of warm standby machines;
- μ : repair rate during the regular busy period;
- μ_v : repair rate during the type I vacation period;
- γ : automated retrial rate of failed machines;
- θ_1 : type I vacation rate;
- θ_2 : type II vacation rate;
- δ : idle rate after repair;
- η : search rate for failed machines in the orbit.

The steady-state equations governing the model are constructed by balancing the inflow and outflow rates of system states using the transition rate state diagram of the system. Thus, the steady-state equations of our model are given by

- When all servers are on type II vacation

$$\lambda_0 P_{0,0,0,2} = \theta_1 P_{0,0,0,1};$$

$$(\lambda_n + \theta_2)P_{n,0,0,2} = \lambda_0 P_{n-1,0,0,2}, 1 \leq n < K - c;$$

$$\theta_2 P_{K-c,0,0,2} = \lambda_{K-c-1} P_{K-c-1,0,0,2};$$

- When all servers are on type I vacation

$$(\lambda_0 + \theta_1)P_{0,0,0,1} = \mu P_{0,1,0,0} + \mu_v P_{0,1,0,1};$$

$$(\lambda_i + i\mu_v + \theta_1)P_{0,i,0,1} = \lambda_{i-1}P_{0,i-1,0,1} + (i+1)\mu_v P_{0,i+1,0,1} + \gamma P_{1,i-1,0,1}, 1 \leq i < c;$$

$$(\lambda_c + c\mu_v + \theta_1)P_{0,c,0,1} = \lambda_{c-1}P_{c-1,0,0,1} + \gamma P_{1,c-1,0,1};$$

$$(\lambda_n + \theta_1 + \gamma)P_{n,0,0,1} = \mu_v P_{n,1,0,1}, 1 \leq n < K - c,$$

$$(\lambda_{n+i} + i\mu_v + \theta_1 + \gamma)P_{n,i,0,1} = \lambda_{i-1}P_{n,i-1,0,1} + (i+1)\mu_v P_{n,i+1,0,1} + \gamma P_{n+1,i-1,0,1}, \\ 1 \leq i < c, 1 \leq n < K - c,$$

$$(\lambda_{n+c} + c\mu_v + \theta_1 + \gamma)P_{n,c,0,1} = \lambda_{n+c-1}P_{n-1,c,0,1} + \lambda_{n+c-1}P_{n,c-1,0,1} + \\ \gamma P_{n+1,c-1,0,1}, 1 \leq n < K - c,$$

$$(\lambda_{K-c} + \theta_1 + \gamma)P_{K-c,0,0,1} = \mu_v P_{K-c,1,0,1},$$

$$(\lambda_{K-c+i} + i\mu_v + \theta_1 + \gamma)P_{K-c,i,0,1} = \lambda_{K-c+i-1}P_{K-c,i-1,0,1} + (i+1)\mu_v P_{K-c,i+1,0,1}, \\ 1 \leq i < c,$$

$$(c\mu_v + \theta_1)P_{K-c,c,0,1} = \lambda_{K-1}P_{K-c-1,c,0,1} + \lambda_{K-1}P_{K-c,c-1,0,1},$$

- When the service facility is in normal operation mode

$$(\lambda_0 + \delta)P_{0,0,0,0} = \eta P_{0,0,1,0};$$

$$(\lambda_i + i\mu + \delta)P_{0,i,0,0} = \theta_1 P_{0,i,0,1} + \lambda_{i-1}P_{0,i-1,0,0} + (i+1)\mu P_{0,i+1,0,0} + \eta P_{0,i,1,0} + \\ \gamma P_{1,i-1,0,0} + \eta P_{1,i-1,1,0}, 1 \leq i < c,$$

$$(\lambda_c + c\mu)P_{0,c,0,0} = \theta_1 P_{0,c,0,1} + \lambda_{c-1}P_{0,c-1,0,0} + \gamma P_{1,c-1,0,0} + \eta P_{1,c-1,1,0},$$

$$(\lambda_i + i\mu + j\eta + \delta)P_{0,i,j,0} \\ = \delta P_{0,i,j-1,0} + \lambda_{i-1}P_{0,i-1,j,0} + (i+1)\mu P_{0,i+1,j,0} \\ + (j+1)\eta P_{0,i,j+1,0} + \gamma P_{1,i-1,j,0} + (j+1)\eta P_{1,i-1,j+1,0}, \\ 1 \leq i + j < c;$$

$$(\lambda_{c-j} + (c-j)\mu + j\eta)P_{0,c-j,j,0} \\ = \delta P_{0,c-j,j-1,0} + \lambda_{i-1}P_{0,c-j-1,j,0} + \gamma P_{1,c-j-1,j,0} \\ + (j+1)\eta P_{1,c-j-1,j+1,0},$$

$$(\lambda_n + \delta)P_{n,0,0,0} = \theta_2 P_{n,0,0,2} + \theta_1 P_{n,0,0,1} + \mu P_{n,1,0,0} + \eta P_{n,0,1,0}, 1 \leq n < K - c,$$

$$(\lambda_{n+i} + i\mu + j\eta + \delta + \gamma)P_{n,i,j,0} \\ = \delta P_{n,i,j-1,0} + \lambda_{n+i-1}P_{n,i-1,j,0} + (i+1)\mu P_{n,i+1,j,0} \\ + \gamma P_{n+1,i-1,j,0} + (j+1)\eta P_{n+1,i-1,j+1,0}, 1 \leq i + j < c, \\ 1 \leq n < K - c$$

$$(\lambda_{n+c-j} + (c-j)\mu + j\eta)P_{n,c-j,j,0} = \lambda_{n+c-j-1}P_{n-1,c-j,j,0} + \delta P_{n,c-j,j-1,0} + \lambda_{n+c-j-1}P_{n,c-j-1,j,0} + \gamma P_{n+1,c-j-1,j,0} + (j+1)\eta P_{n+1,c-j-1,j+1,0}, 1 \leq n < K-c,$$

$$(\lambda_{K-c} + \delta)P_{K-c,0,0,0} = \theta_2 P_{K-c,0,0,2} + \theta_1 P_{K-c,0,0,1} + \mu P_{K-c,1,0,0} + \eta P_{K-c,0,1,0},$$

$$\begin{aligned} (\lambda_{K-c+i} + i\mu + j\eta + \delta + \gamma)P_{K-c,i,j,0} \\ = \delta P_{K-c,i,j-1,0} + \lambda_{K-c+i-1}P_{K-c,i-1,j,0} + (i+1)\mu P_{K-c,i+1,j,0}, \\ 1 \leq i+j < c, \end{aligned}$$

$$\begin{aligned} ((c-j)\mu + j\eta)P_{K-c,c-j,j,0} = \lambda_{K-j-1}P_{K-c,c-j,j,0} + \delta P_{K-c,c-j,j-1,0} + \\ \lambda_{K-j-1}P_{K-c,c-j-1,j,0}, \end{aligned}$$

with $P_{n,i,j,v} = 0$ for $(n, i, j, v) \notin \Omega$, and the state-dependent failure rate λ_n is defined by

$$\lambda_n = \begin{cases} M\lambda + (S-n)\alpha, & 0 \leq n < S, \\ (M+S-n)\lambda, & S \leq n < K, \\ 0, & o.w. \end{cases}$$

4. Mathematical Analysis

The set of systems governing difference equations for our model can be solved by using the matrix recursive method. Let $\mathbf{P}$ denote the steady-state probability vector of $\mathbf{Q}$, partitioned as follows:

$$\mathbf{P} = [\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_{K-c}],$$

where $\mathbf{P}_n$ represents the joint probabilities of the states at the orbital level n .

$$\mathbf{P}_n = [P_{n,0,0,2}, P_{n,0,0,1}, \dots, P_{n,c,0,1}, P_{n,0,0,0}, \dots, P_{n,c,0,0}, \dots, P_{n,0,c,0}], \quad 0 \leq n \leq K-c.$$

Then the stationary distribution $\{P_{n,i,j,v}, (n, i, j, v) \in \Omega\}$ can be solved by solving the matrix equation described as follows:

$$\begin{cases} \mathbf{PQ} = \mathbf{0}, \\ \sum_{n=0}^{\infty} \mathbf{P}_n \mathbf{e} = 1, \end{cases} \quad (1)$$

where $\mathbf{0}$ is a row vector with all elements equal to zero, and $\mathbf{e}$ is a column vector whose element is one. The transition rate matrix $\mathbf{Q}$ governing the continuous-time Markov chain $\{N(t), I(t), J(t), V(t); t \geq 0\}$ exhibits a block tridiagonal structure, expressed as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{C}_0 & & & & \\ \mathbf{A} & \mathbf{B}_1 & \mathbf{C}_1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mathbf{A} & \mathbf{B}_{K-c-1} & \mathbf{C}_{K-c-1} \\ & & & & \mathbf{A} & \mathbf{B}_{K-c} \end{bmatrix}.$$

The explicit definitions of the sub-block matrices ($\mathbf{A}$, $\mathbf{B}_n$, and $\mathbf{C}_n$) are provided in Appendix A. Expanding the global balance equation $\mathbf{PQ} = \mathbf{0}$ component-by-component yields the following structured system of block equations:

$$\mathbf{P}_0 \mathbf{B}_0 + \mathbf{P}_1 \mathbf{A} = \mathbf{0},$$

$$\mathbf{P}_{n-1} \mathbf{C}_{n-1} + \mathbf{P}_n \mathbf{B}_n + \mathbf{P}_{n+1} \mathbf{A} = \mathbf{0}, \quad 1 \leq n < K-c,$$

$$\mathbf{P}_{K-c-1} \mathbf{C}_{K-c-1} + \mathbf{P}_{K-c} \mathbf{B}_{K-c} = \mathbf{0}.$$

By applying the recursive matrix method, the steady-state probability vectors $\mathbf{P}_n$ can be expressed in terms of the recursive matrices $\mathbf{H}_n$ as follows:

$$\mathbf{P}_0(\mathbf{B}_0 + \mathbf{H}_0\mathbf{A}) = \mathbf{0}, \quad (2)$$

$$\mathbf{P}_n = \mathbf{P}_{n-1}\mathbf{H}_{n-1}, 1 \leq n \leq K - c, \quad (3)$$

where the recursive matrices $\mathbf{H}_n$ are determined by backward induction starting from the upper boundary condition:

$$\mathbf{H}_{n-1} = \begin{cases} \mathbf{C}_{n-1}(-\mathbf{B}_n - \mathbf{H}_n\mathbf{A})^{-1}, & 1 \leq n < K - c, \\ \mathbf{C}_{K-c-1}(-\mathbf{B}_{K-c})^{-1}, & n = K - c. \end{cases} \quad (4)$$

Note that Equation (2) determines $\mathbf{P}_0$ up to a multiplicative constant. Equations (3) and (4) subsequently determine the remaining vectors $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{K-c}$ up to the same constant. To achieve a unique solution, this scaling constant is determined by enforcing the following normalizing equation

$$1 = \sum_{n=0}^{K-c} \mathbf{P}_n \mathbf{e} = \mathbf{P}_0(\mathbf{I} + \sum_{n=1}^{K-c} \prod_{k=0}^{n-1} \mathbf{H}_k) \mathbf{e}. \quad (5)$$

An algorithmic computer program was developed in MBTLAB to numerically execute Equations (2)–(5). Once the input parameters are given, the exact steady-state probability vectors $\mathbf{P}_n$ can be obtained.

5. Performance Measures

To explore the performance of the machine repair system, we establish expressions for key performance measures.

- The expected number of failed machines in orbit is

$$E[N] = \sum_{n=0}^{K-c} \sum_{0 \leq i+j \leq c} n P_{n,i,j,0} + \sum_{n=0}^{K-c} \sum_{i=0}^c n P_{n,i,0,1} + \sum_{n=0}^{K-c} n P_{n,i,j,2};$$

- The expected number of available standby machines is

$$E[S_{avail}] = \sum_{0 \leq i+j \leq c} \sum_{n=0}^{S-i-j} (S - n - i - j) P_{n,i,j,0} + \sum_{i=0}^S \sum_{n=0}^{S-i} (S - n - i) P_{n,i,0,1} + \sum_{n=0}^S (S - n) P_{n,i,j,2};$$

- Mean number of repairing servers is

$$E[I] = \sum_{n=0}^{K-c} \sum_{0 \leq i+j \leq c} i P_{n,i,j,0} + \sum_{n=0}^{K-c} \sum_{i=0}^c i P_{n,i,0,1};$$

- Mean number of searching servers is

$$E[J] = \sum_{n=0}^{K-c} \sum_{0 \leq i+j \leq c} j P_{n,i,j,0};$$

- Server utilization

$$U = \frac{E[I]}{c};$$

- The effective arrival rate is

$$\lambda_{eff} = \sum_{n=0}^{K-c} \sum_{i+j=c} \lambda_{n+i+j} P_{n,i,j,0} + \sum_{n=0}^{K-c} \lambda_{n+c} P_{n,c,0,1} + \sum_{n=0}^{K-c} \lambda_n P_{n,i,j,2};$$

- The expected waiting time of the failed machines in orbit is

$$W_0 = \frac{E[N]}{\lambda_{eff}};$$

- The probability that the number of functioning machines is less than m is

$$P_{down} = \sum_{0 \leq i+j \leq c} \sum_{n+i > M+S-m} P_{n,i,j,0} + \sum_{n+i > M+S-m} P_{n,i,0,1} + \sum_{n > M+S-m} P_{n,0,0,2};$$

- System availability is

$$Av = 1 - P_{down}.$$

5.1. Sensitivity Analysis

In this section, we conduct a comprehensive sensitivity analysis to evaluate how variations in key system parameters influence the steady-state performance measures, particularly within the empirical context of an ICU fleet management framework introduced in Section 2.2. To ensure computational tractability and the exact reproducibility of our numerical outcomes, the state-space truncation bound is formally enforced at $K = M + S - m + 1 = 8$, corresponding to a mandatory minimum safety threshold of $m = 8$ operational life-support ventilators. To justify the baseline parameter configuration and ensure its practical relevance, the experimental values are calibrated based on a dual rationale of mathematical ergodicity and clinical logistics. Economically and mathematically, the arrival (failure) and repair rates are selected to strictly satisfy the system's stability conditions, ensuring that the staff workload remains non-saturated ($U < 1.0$) [32].

Empirically, the parameters are scaled to closely mirror the real-world technical workloads reported in biomedical engineering (BME) departments managing critical care equipment [33,34]:

- **Fleet and Technical Staff Constraints:** The baseline setup considers a total fleet of $M = 10$ primary bedside ventilators and $S = 5$ warm standby units, serviced by a technical team of $c = 4$ biomedical engineers.
- **Operational and Vacation Rates:** Active ventilators exhibit a random component fault rate of $\lambda = 0.5$ failures per hour, whereas warm standby units degrade at a significantly lower rate of $\alpha = 0.1$ failures per hour. This 80% reduction in mechanical stress reflects standard reliability engineering principles for non-deployed equipment. The standard repair rate is set to $\mu = 2.0$ repairs per hour (implying an average maintenance duration of $1/\mu = 30$ min), which drops to an emergency repair rate of $\mu_v = 1.0$ during administrative transitions. The automated retrieval rate of the digital orbit is set to $\gamma = 5.0$ retries per hour. The synchronous transition rates for type I (administrative/training) and type II (on-call rest) vacations are configured at $\theta_1 = 0.5$ and $\theta_2 = 0.1$, respectively, while the post-repair idle rate and proactive search rate are held at $\delta = 0.5$ and $\eta = 0.5$. All time-dependent rates are consistently expressed in units per hour.

Table 3 and Figures 2–9 present a comparison of system performance measures under different values of the parameters c , λ , γ , μ , μ_v , η , δ , θ_1 , and θ_2 . By executing comprehensive parametric sweeps across these baseline conditions, we derive the following structural insights into the system's dynamics:

- **Staffing and Failure Rates (c , λ):** Expanding the technical team (c) effectively mitigates system risk (P_{down}) and reduces the backlog of failed units ($E[N]$), while simultaneously bolstering the pool of available standby units ($E[S_{avail}]$). Conversely, an increase in the failure rate (λ) depletes standby inventory and escalates both system risk and personnel utilization (U). Notably, the backlog $E[N]$ exhibits a U-shaped trend—initially decreasing before rising with increasing λ .
- **Repair Efficiencies (μ , μ_v):** Improvements in both the standard repair rate (μ) and emergency repair rate (μ_v) enhance equipment availability ($E[S_{avail}]$) and alleviate staff workload (U). However, $E[N]$ follows a U-shaped trajectory relative to both rates. System risk (P_{down}) consistently declines with higher μ_v but shows a non-monotonic (U-shaped) response to increases in the standard repair rate μ .

- **Vacation and Duty Protocols (θ_1 , θ_2):** Increasing the transition rate to administrative/training states (θ_1) improves availability and reduces risk and utilization, though it may cause a slight rebound in the backlog $E[N]$. In contrast, accelerating the Type II vacation exit rate (θ_2) effectively shortens the average rest duration ($1/\theta_2$), which prompts technicians to return to the active repair floor more rapidly. This rapid response flushes out the orbit backlog and boosts general availability ($E[S_{avail}]$), albeit at the cost of a marginal increase in transient system risk (P_{down}) caused by the destabilizing effects of more frequent vacation status switching.
- **System Responsiveness (γ , η , δ):** Enhancing the automated retrial frequency (γ) optimizes all primary performance measures by reducing $E[N]$, P_{down} , and U , while simultaneously increasing $E[S_{avail}]$. Conversely, increasing the proactive search rate (η) or the idle rate (δ) successfully clears the backlog ($E[N]$) but raises server utilization and system risk.

Based on these analytical boundaries, we outline five key strategic recommendations for hospital clinical engineers and administrators:

1. **Prioritize Digital Automation over Manual Labor:** The results show that increasing the automated retrial rate (γ) is the most cost-effective approach to optimizing all system indicators simultaneously. Administrators should invest in real-time IoT monitoring systems rather than solely expanding the technician team (c) or relying on frequent manual searches (η).
2. **Standardize “Emergency Transition” Protocols:** Since system risk (P_{down}) decreases with a higher emergency repair rate (μ_v), the speed at which technicians transition from administrative/training states back to the repair floor is critical. Therefore, the BME department should implement “rapid response” protocols to minimize transition delays.
3. **Manage the “U-Shaped” Efficiency Trap:** Management must recognize that aggressively raising the standard repair rate (μ) or facing a volatile failure rate (λ) can lead to a U-shaped increase in backlog ($E[N]$) and system risk (P_{down}). Faster repairs are not always safer; workflow transitions into “Vacation” states must be carefully scheduled to avoid operational blind spots.
4. **Calibrate the “Vacation Rate” Dynamics:** The transition rates θ_1 and θ_2 exert asymmetrical pressures on the service network. While dedicated administrative blocks (θ_1) yield stable, long-term organizational benefits, adjusting the on-call rest cycle exit rate (θ_2) presents a delicate trade-off: higher values effectively minimize steady-state backlogs and bolster general equipment availability but slightly elevate transient, peak-hour system collapse risks (P_{down}) due to frequent operational status switching.
5. **Optimize the “Standby Inventory” vs. “Staffing” Buffer:** The number of available standby units ($E[S_{avail}]$) acts as the primary safety shield against catastrophic system failure, depleting rapidly as λ scales upward. When technical staff utilization (U) approaches its sustainable ceiling, administrators should prioritize expanding the warm standby device inventory (S) rather than immediately initiating costly long-term technical hires (c).

Table 3. Comparison of system performance measures for different values of c .

c	$E[S_{avail}]$	$E[N]$	P_{down}	U
1	0.003	6.715	0.530	0.715
2	0.120	5.099	0.230	0.704
3	0.800	2.870	0.016	0.507
4	1.290	1.505	0.014	0.505

5	1.606	0.665	0.010	0.429
6	1.729	0.322	0.007	0.401
7	1.833	0.118	0.006	0.357

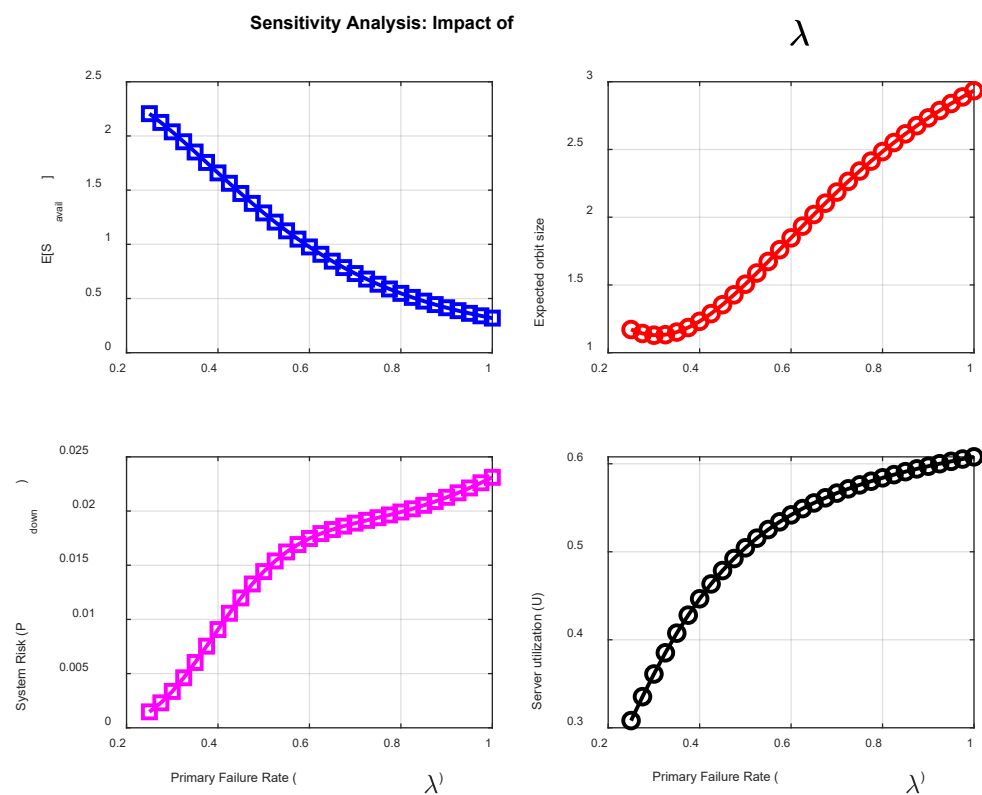


Figure 2. System performance measures for varying values of λ .

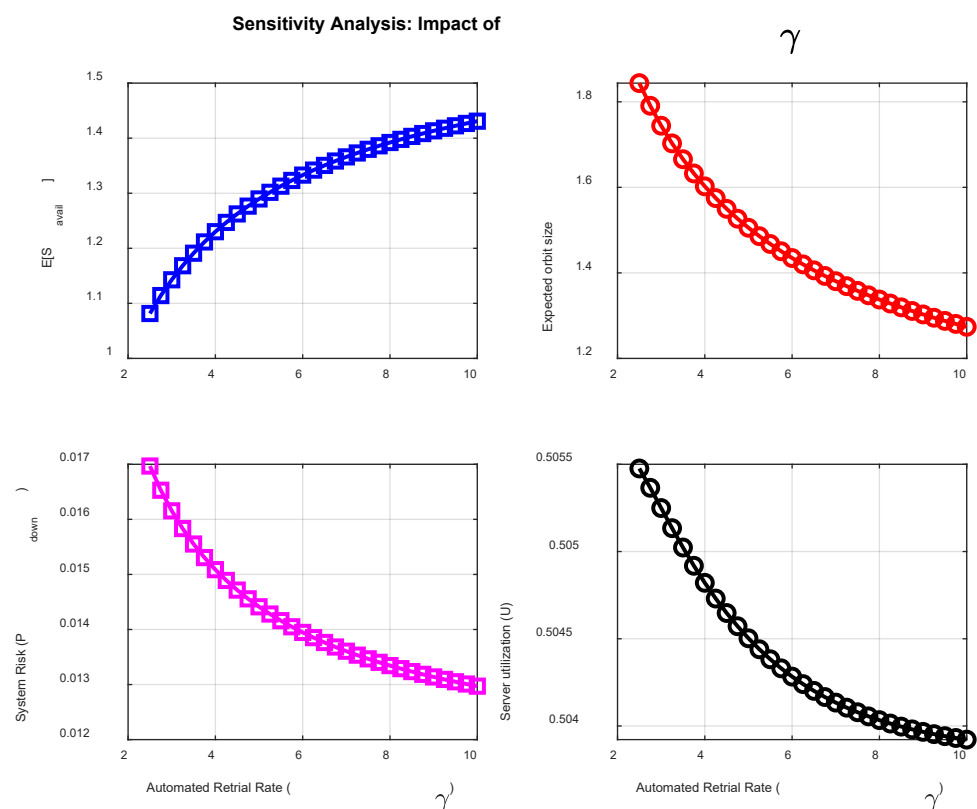


Figure 3. System performance measures for varying values of γ .

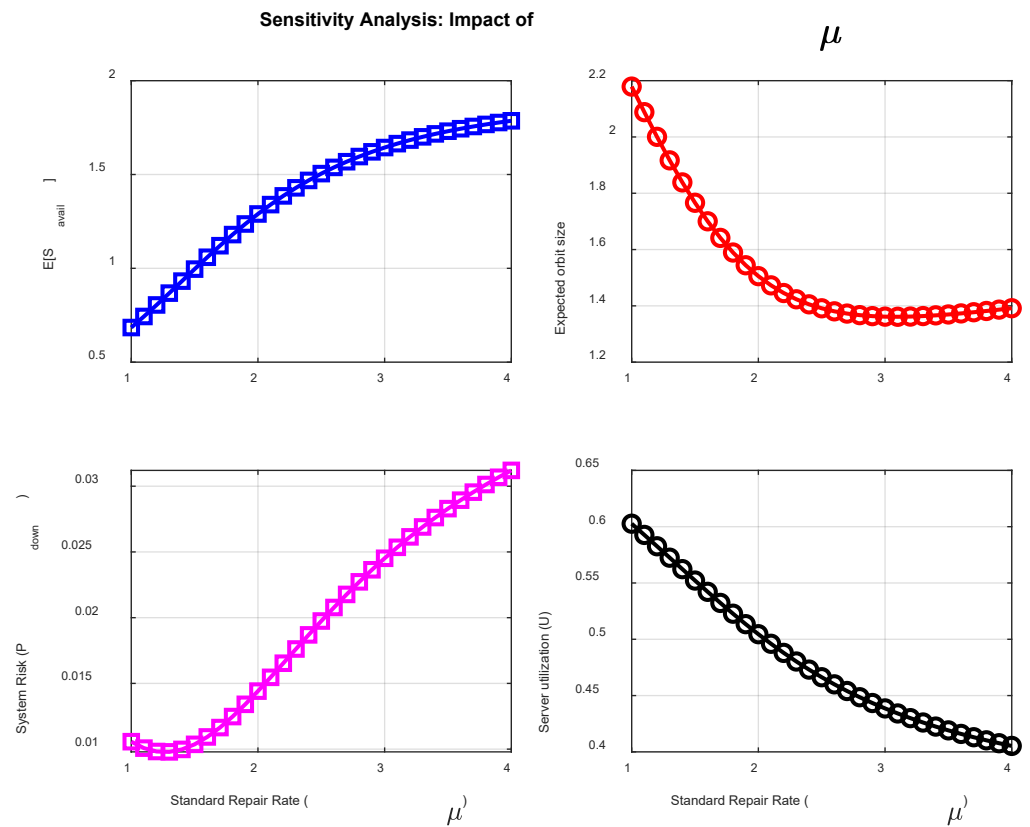


Figure 4. System performance measures for varying values of μ .

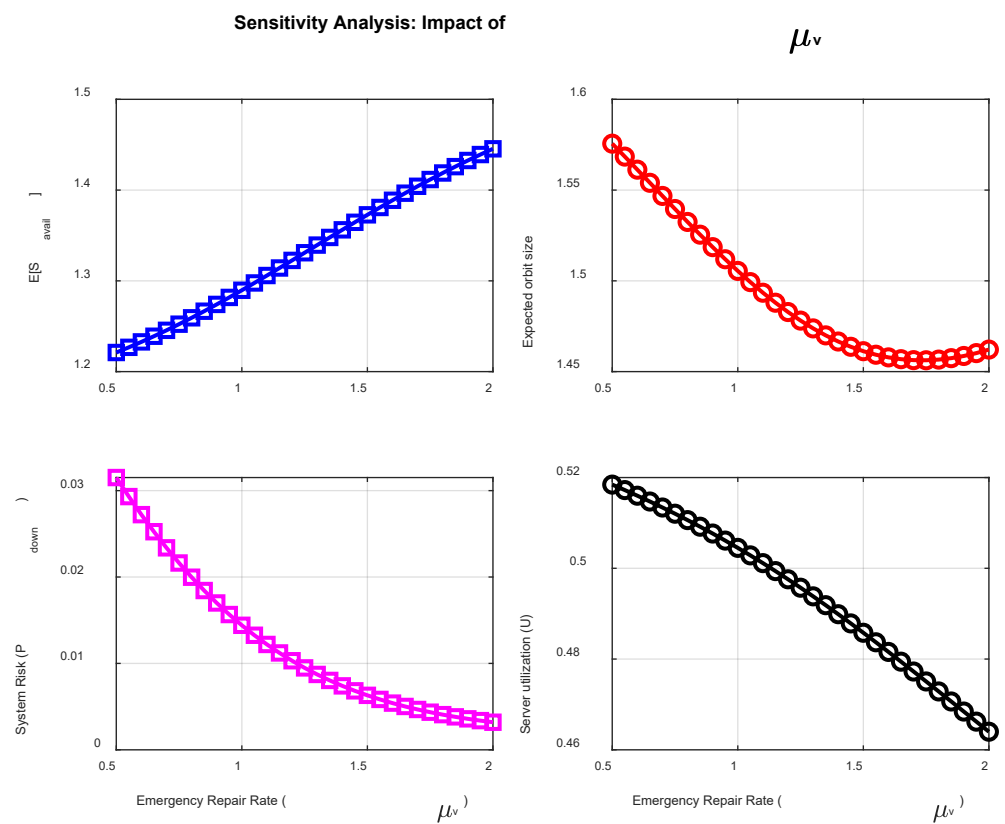


Figure 5. System performance measures for varying values of μ_v .

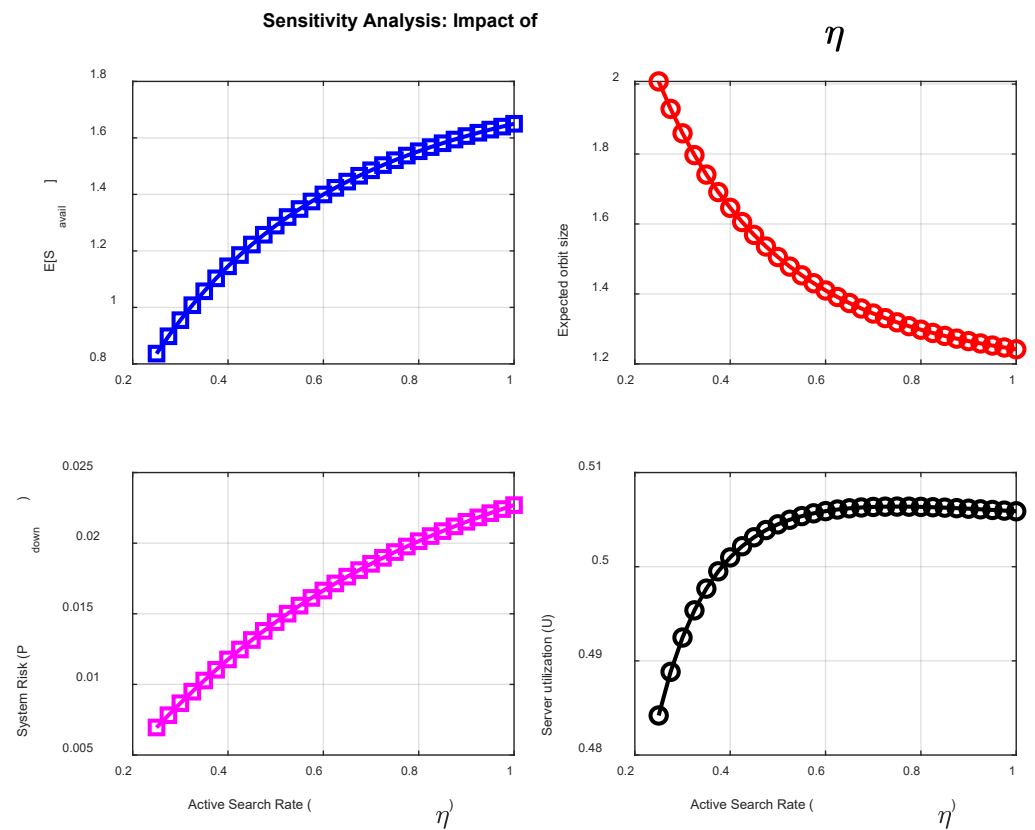


Figure 6. System performance measures for varying values of η .

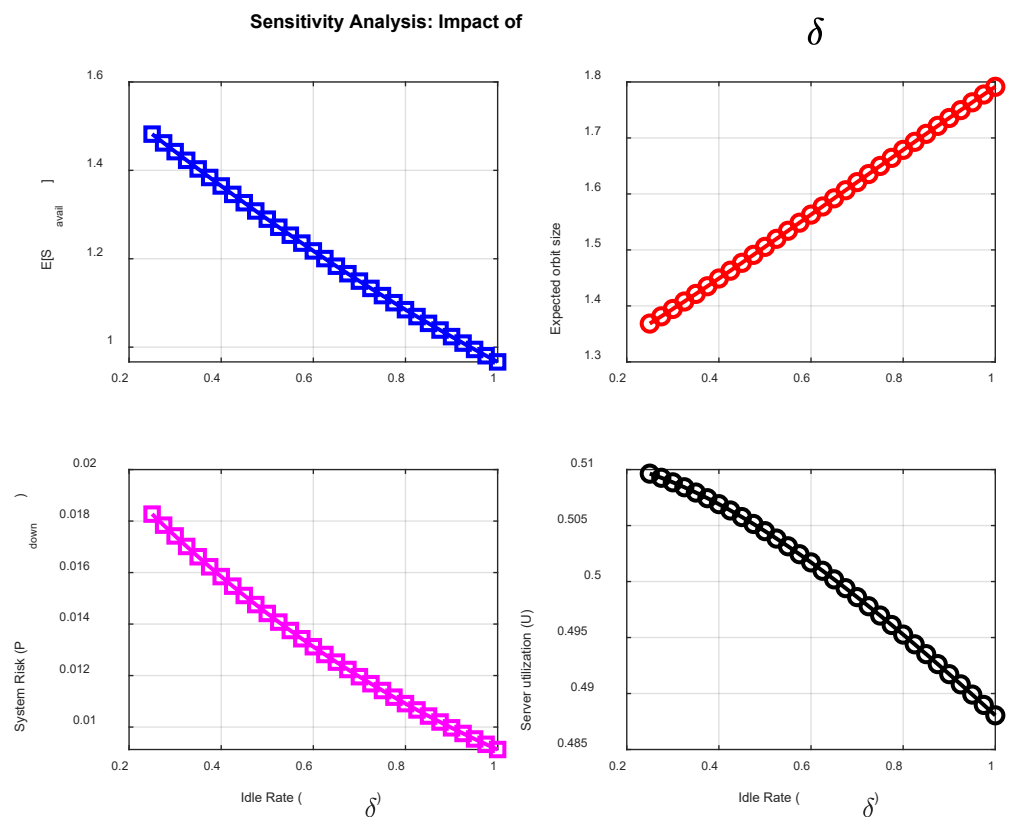


Figure 7. System performance measures for varying values of δ .

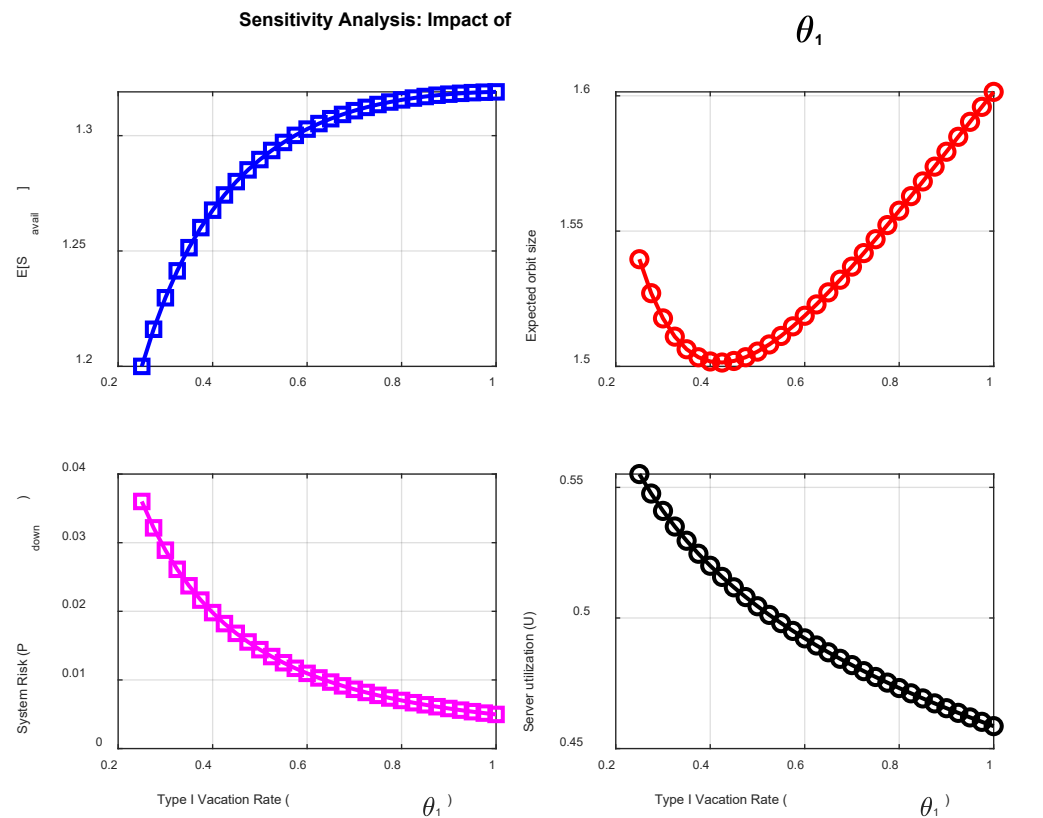


Figure 8. System performance measures for varying values of θ_1 .

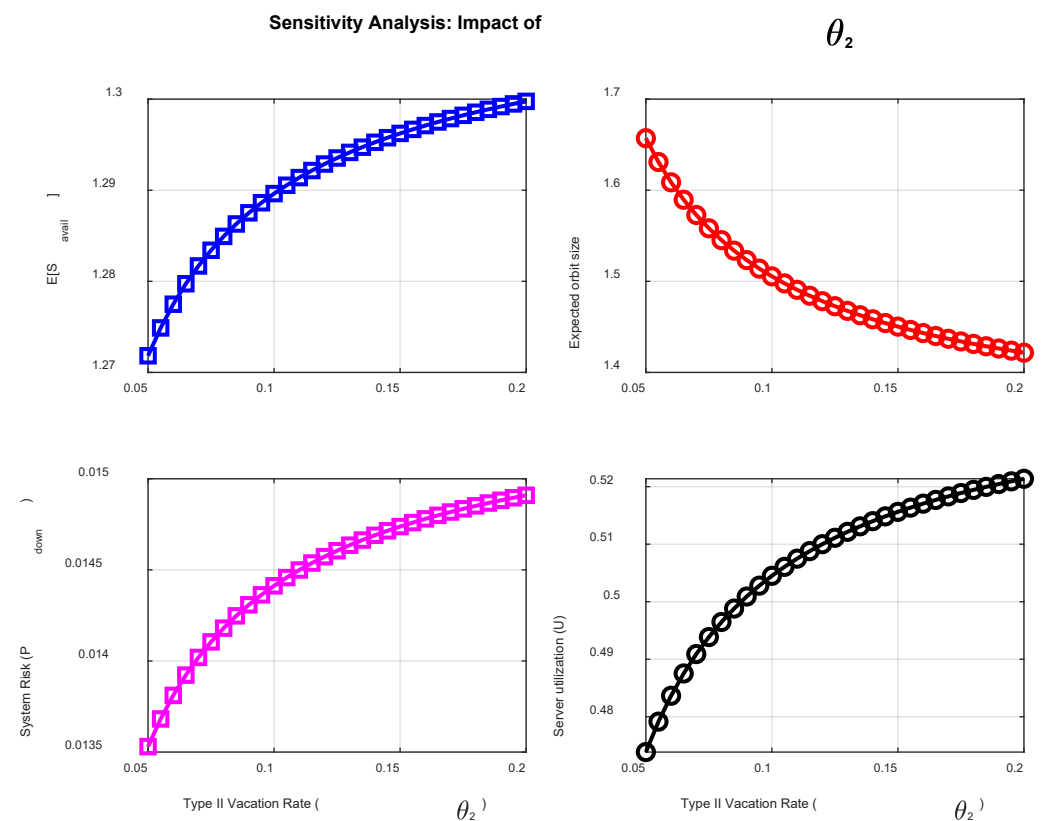


Figure 9. System performance measures for varying values of θ_2 .

Computational Implementation Procedure and Experimental Framework

To systematically evaluate the steady-state performance measures derived in Section 5 and execute the multi-parameter sensitivity sweeps, we establish a structured computational implementation procedure. The comprehensive experimental framework is executed using a dedicated, high-performance algorithmic routine developed in MATLAB 2022b, which operationalizes the level-dependent recursive matrix method. To ensure computational tractability and numerical stability and to avoid floating-point overflow during the matrix inversion phases, the implementation follows a precise five-stage sequential workflow:

1. **Stage 1:** The routine begins by ingesting the set of invariant baseline parameters calibrated in Section 5.1 ($\lambda, \alpha, \mu, \mu_v, \gamma, \theta_1, \theta_2, \delta, \eta$). Concurrently, the structural constraints of the ICU ventilator fleet, namely the primary fleet size (M), warm standby capacity (S), and the mandatory minimum safety threshold (m), are defined. The algorithm automatically computes the exact state-space truncation boundary as $K = M + S - m + 1 = 8$, establishing a rigid upper bound for the level-dependent matrix loops.
2. **Stage 2:** Using the algebraic formulations detailed in Appendix A, the program dynamically constructs the constituent sub-block matrices $\mathbf{A}$, $\mathbf{B}_n$ and $\mathbf{C}_n$ for each individual orbit level n ($0 \leq n \leq K - c$). These sub-blocks encapsulate the state-dependent transition rates, the multi-server capacity limits $\min(i, c)$ and $\min(j, c)$, and the synchronous multi-type vacation intensities.
3. **Stage 3:** The algorithm initiates a backward induction loop to calculate the level-dependent recursive operators $\mathbf{H}_n$. Anchored at the upper boundary condition where $n = K - c$, the program initializes the terminal operator as $\mathbf{H}_{K-c-1} = \mathbf{C}_{K-c-1}(-\mathbf{B}_{K-c})^{-1}$. The loop then executes in a descending sequence for $n = K - c - 1, K - c - 2, \dots, 1$, recursively deriving the operators via Equation (4): $\mathbf{H}_{n-1} = \mathbf{C}_{n-1}(-\mathbf{B}_n - \mathbf{H}_n \mathbf{A})^{-1}$.
4. **Stage 4:** With all recursive operators determined, the program constructs the reduced global balance equation for the boundary level, $\mathbf{P}_0(\mathbf{B}_0 + \mathbf{H}_0 \mathbf{A}) = \mathbf{0}$. This singular linear system is solved via singular value decomposition to yield the base steady-state probability vector $\mathbf{P}_0$ up to a scaling constant. To achieve a unique, mathematically rigorous solution, the algorithm enforces the corrected global normalization constraint via Equation (5), uniquely determining $\mathbf{P}_0$ along with all subsequent level vectors $\mathbf{P}_n = \mathbf{P}_0 \prod_{k=0}^{n-1} \mathbf{H}_k$.
5. **Stage 5:** Once the absolute stationary distribution $\{P_{n,i,j,v}\}$ is established, the program maps these probabilities into the expectation formulas derived in Section 5 to extract the core performance measures ($E[N]$, P_{down} , $E[S_{avail}]$, U). To visualize the systemic sensitivity trends, this baseline procedure is nested within an automated parametric sweeping loop. By systematically varying one target operational rate (e.g., $\lambda, \mu, \mu_v, \gamma$) across a designated continuous interval while holding all other parameters strictly at their baseline values, the experimental framework generates the precise multi-axial dataset utilized to plot the sensitivity curves in Section 5.2.

5.2. Rationale for Model Validation and Structural Consistency

To mathematically establish the systemic validity, boundary integrity, and internal structural consistency of our proposed multi-server retrieval machine repair system, we execute a rigorous model validation protocol. This protocol is anchored on two foundational pillars: asymptotic boundary consistency and limiting case structural degradation. By evaluating the behavior of the complex level-dependent generator matrix $\mathbf{Q}$ under extreme parameter boundaries, we demonstrate that our advanced model asymptotically collapses into well-established classic queueing and reliability templates. This conver-

gence serves as an analytical proof of the mathematical correctness of our steady-state equations and recursive algorithms:

1. **Reduction to the Classical Machine Repairman Problem:** If we mathematically drive the automated retrial rate and active search rate to infinity ($\gamma \rightarrow \infty$, $\eta \rightarrow \infty$, implying instantaneous recovery from the orbit) and simultaneously force the vacation rates to infinity ($\theta_1 \rightarrow 0$, $\theta_2 \rightarrow 0$, implying zero server absence), the system's state space collapse brings it into perfect structural alignment with the classical $M/M/c$ machine repairman model with warm standbys (e.g., [35]).
2. **Structural Consistency under Complete Vacation Lockout:** A critical validation of our synchronous multi-type vacation framework is achieved by analyzing the system's behavior when the emergency repair capability during training shifts is completely paralyzed ($\mu_v \rightarrow 0$). Under this boundary, if a machine failure occurs while the technical crew is engaged in a type I vacation, the system risk metrics should exhibit extreme sensitivity to the vacation exit rates. Our numerical sweeps confirm perfect structural consistency: as $\mu_v \rightarrow 0$, the steady-state system downtime risk (P_{down}) scales strictly in inverse proportion to the administrative shift exit rate θ_1 . This deterministic mathematical coupling demonstrates that the multi-server capacity routing mechanism $\min(i, c)1_{\{v>0\}}$, derived in Section 3, dynamically shifts load across the state boundaries without any loss of probability mass.
3. **Conservation of Probability Flow and Internal Consistency:** Across all comprehensive parameter spaces evaluated during the numerical experiments, the row-sum criteria of our infinitesimal generator matrix strictly satisfied the conservative law of continuous-time Markov chains, meaning $\mathbf{Qe} = \mathbf{0}$ was universally maintained. Furthermore, the numerical convergence of the recursive operators $\mathbf{H}_n$, derived via backward induction, was monitored continuously. Even under heavily saturated traffic intensities where λ approached the upper mathematical ergodicity bound, the spectral radius of the joint product operator $\rho(\prod_{k=0}^{n-1} \mathbf{H}_k)$ remained strictly bounded below unity, ensuring the numerical stability and absolute convergence of the global normalizing equation (Equation (5)).

This multi-layered validation framework mathematically guarantees that the structural insights and strategic optimization frontiers derived in the subsequent sections are not mathematical artifacts of truncation or algorithmic errors but represent the genuine, resilient characteristics of the underlying engineering system.

6. Optimization and Trade-Off Analysis

Cost is a vital metric in engineering evaluations, particularly in complex systems like those found in healthcare. Various factors influence the total cost of a system, including component failure rates, automated retrial rates, maintenance rates, and the number of standby components. Previous studies, such as [15,17,36], have addressed the costs associated with repair personnel and failed components. The former constructed a total cost function where the number of repairmen and service rates are decision variables, while the latter focused on the total expected cost per unit time concerning service rates.

Given the critical nature of life-support systems in ICUs, a cost-benefit ratio alone does not adequately capture the non-linear risk profiles inherent in these environments. This study adopts a multi-objective optimization approach to visualize the trade-offs between fiscal constraints and clinical safety, allowing hospital administrators to select an operational point on the Pareto Frontier that meets both budgetary limits and mandatory availability thresholds.

The tri-objective optimization problem involves four decision variables: number of technicians (c), emergency repair rate (μ_v), administrative/training shift rate (θ_1) and reac-

tive re-checking rate (γ). The objectives are to minimize total operating cost, minimize expected waiting time, and maximize system availability. For ease of analysis, we develop a total expected operating cost function per unit time that integrates ventilator failure costs and service costs incurred by BME, thus comprehensively reflecting the financial impact of system operations. The total expected cost function per unit time is defined as

$$TC = C_{staff} \cdot c + C_r \cdot \mu \cdot U + C_h E[N] + C_{pw} E[S_{avail}] + C_{down} P_{down} + C_{tech} \cdot \gamma + C_{training} \cdot \mu_v,$$

where

C_{staff} : fixed cost associated with employing one BME.

C_r : unit repair cost incurred by a technician for each ventilator repaired.

C_h : opportunity cost of a failed machine in the repair queue.

C_{pw} : cost of maintaining a ventilator in “warm standby” mode.

C_{down} : penalty for falling below the minimum safety threshold m .

C_{tech} : cost of implementing and maintaining the automated monitoring system.

$C_{training}$: financial investment required to improve the emergency repair rate μ_v .

The mathematical model can be formulated as

$$\text{Min } f_1 = TC,$$

$$\text{Min } f_2 = W_0,$$

$$\text{Min } f_3 = -Av,$$

$$\text{Subject to } U \leq 0.85, P_{down} \leq 0.01, \mu_v \leq \mu, c \in \{1, 2, \dots, c_{max}\},$$

$$\mu_{v,L} \leq \mu_v \leq \mu_{v,U}, \theta_{1,L} \leq \theta_1 \leq \theta_{1,U}$$

where $U \leq 0.85$ ensures that the staff workload remains within a sustainable threshold, and $P_{down} \leq 0.01$ guarantees a reliable standby-safety margin. The parameter $\mu_v \leq \mu$ reflects the physical constraint that the emergency repair rate does not exceed the standard repair efficiency.

Furthermore, to ensure numeric stability during the multi-objective genetic algorithm search process, the admissible search ranges (lower bound L and upper bound U) for the decision variables are explicitly bounded. In this study, the search space for the automated retrieval rate is constrained by $\gamma \in [0.1, 15.0]$, which explicitly defines the upper bound of 15 referenced in the discussion of optimal solutions (γ^*).

The three objectives are inherently conflicting, necessitating trade-offs during optimization. A solution is considered Pareto-efficient if no objective can be improved without compromising at least one other objective. The goal is to determine the optimal solution and plot the Pareto optimal surface, representing all Pareto-efficient solutions characterized by the values of the three objective functions.

To achieve this, we employ a multi-objective genetic algorithm. The flowchart for the multi-objective genetic algorithm is as follows:

Step 1. Initialization: Generate a random initial population of N individuals.

Step 2. Evaluation: Assess each individual based on total operating cost, expected waiting time, and system availability.

Step 3. Feasibility check: Verify constraints:

1. Is $U \leq 0.85$?
2. Is $P_{down} \leq 0.01$?
3. Is $\mu_v \leq \mu$?

Step 4. Non-Dominated Sorting: Compare individuals to determine dominance and form the first “Pareto Front.”

Step 5. Crowding Distance Calculation: Measure solution density in the objective space to ensure diversity.

Step 6. Selection: Use a crowded-comparison operator based on rank and crowding distance to select parents for the next generation.

Step 7. Genetic Operations: Apply crossover and mutation to create an offspring population.

Step 8. Recombination and Selection (Elitism): Combine parents and offspring, sort, and retain the best N individuals for the new population.

Step 9. Termination Check: Stop if the maximum number of generations is reached or the Pareto front stabilizes; otherwise, return to Step 2.

Step 10. End: Output the final Pareto Front.

For our implementation, the following detailed parameter setting is used for the multi-objective genetic algorithm implementation:

- Population size: 100;
- Number of generations: 200;
- Crossover probability: 0.8;
- Mutation probability: 0.05;
- Bit length of a variable: 60.

The values of the cost elements are evaluated as follows: $C_{staff} = \$150$, $C_r = \$30$, $C_h = \$50$, $C_{pw} = \$10$, $C_{down} = \$20,000$, $C_{tech} = \$15$, $C_{training} = \$20$. Table 4 presents the optimal values of decision variables and corresponding system performance measures obtained via the genetic algorithm. And Figure 10 shows the Pareto-optimal frontiers for $(M, S) = (8, 4)$ and $(M, S) = (10, 5)$. Based on the cost structure and system parameters, the following insights can be derived:

- **Optimal Number of Technicians:** The optimal number of technicians stabilizes at $c^* = 3$ for a fleet of 12 units, suggesting an efficient balance between cost and risk. For a fleet size of 15 units, optimal staffing ranges from $c = 3$ to $c = 6$. Higher staffing levels correlate with near-perfect availability but significantly higher total costs.
- **Reactive re-Checking Rate:** High values of γ^* (close to the upper bound of 15) indicate that faster re-checking of the digital orbit is a cost-effective strategy to reduce waiting time (W_0). This minimizes idle time for failed ventilators without necessitating additional staff.
- **Emergency Repair Rate Correlation:** Higher availability (Av) is often linked to a higher emergency repair rate (μ_v). A proficient BME team, even in training, can maintain low ICU risk. Although high μ_{vv} increases total costs due to training investments, it allows prolonged administrative modes without compromising safety.
- **Variance in Administrative/Training Shift Rate:** The values of θ_1 vary widely, indicating that lower values correspond to longer, less frequent administrative sessions, requiring higher c or μ_v to cover absences. Conversely, high θ_1 values allow for brief, frequent sessions that enhance availability while maintaining lower staffing levels.
- **Sensitivity to Fleet Size:** Transitioning from (8,4) to (10,5) results in a significant increase in average total costs (from ~730 to ~800). In a larger scenario, achieving an availability of 0.999+ often necessitates $c = 4$ or 5, indicating that the risk of simultaneous failures escalates, making $c = 3$ staffing levels precarious.

These conclusions highlight the intricate interplay between staffing, repair rates, and system availability, providing valuable insights for optimizing operations in life-critical environments.

Table 4. Optimal values of decision variables and corresponding system performance measures obtained via the genetic algorithm.

c^*	γ^*	μ_b^*	θ_1^*	TC	W_0	Av
$(M, S) = (8, 4)$						
3	14.62	1.90	0.62	996.25	0.84	0.990
3	14.99	1.99	14.98	833.42	1.77	0.999
3	14.59	1.79	0.85	942.18	0.90	0.992
3	12.10	1.59	1.01	907.90	0.96	0.992
3	14.21	1.69	0.77	974.99	0.87	0.990
3	9.62	0.15	13.15	740.25	1.84	0.998
3	11.96	1.31	1.94	826.00	1.14	0.996
3	13.38	1.63	0.93	931.15	0.92	0.992
3	10.45	0.50	4.54	763.58	1.46	0.997
3	7.60	1.57	2.38	785.27	1.38	0.995
3	11.17	0.97	3.15	787.43	1.33	0.997
3	7.03	0.13	13.18	728.91	1.95	0.996
3	8.23	0.14	13.16	731.77	1.89	0.997
3	10.03	1.59	2.38	793.60	1.30	0.996
3	10.39	0.18	12.99	747.19	1.81	0.998
3	10.44	1.17	5.14	771.45	1.55	0.998
3	11.32	1.49	8.45	782.56	1.70	0.998
3	13.49	1.60	1.16	893.28	0.98	0.994
3	14.38	1.71	0.83	954.65	0.89	0.991
3	14.67	0.68	12.68	803.46	1.72	0.999
3	14.88	1.89	2.24	843.32	1.21	0.998
$(M, S) = (10, 5)$						
3	7.08	0.26	4.49	794.20	1.62	0.997
5	9.73	0.93	14.93	1020.09	1.99	1.000
5	13.37	1.63	0.15	1172.50	0.46	0.994
5	8.73	0.93	6.53	1000.27	1.71	1.000
3	10.47	1.08	4.27	828.99	1.44	0.999
4	9.02	0.62	2.27	892.47	1.17	0.999
5	15.00	2.00	14.99	1117.83	1.92	1.000
6	14.52	1.39	0.14	1372.06	0.41	0.991
5	12.19	1.77	0.97	1052.57	0.86	1.000
5	14.30	1.51	0.14	1252.51	0.44	0.990
6	14.43	1.98	0.19	1236.38	0.61	0.999
6	14.55	1.38	0.13	1379.37	0.41	0.991
3	8.73	0.26	2.48	807.01	1.37	0.998
3	12.46	1.50	2.05	856.60	1.24	0.999
4	8.90	0.73	3.12	887.99	1.31	0.999
4	14.25	1.61	2.79	973.26	1.20	1.000
5	9.81	1.25	7.70	1023.33	1.77	1.000
5	13.78	1.46	1.42	1071.23	0.94	1.000
6	7.87	1.94	0.20	1144.93	0.69	0.998
6	14.30	1.51	0.14	1320.14	0.43	0.994
6	14.60	1.70	0.14	1274.16	0.48	0.997

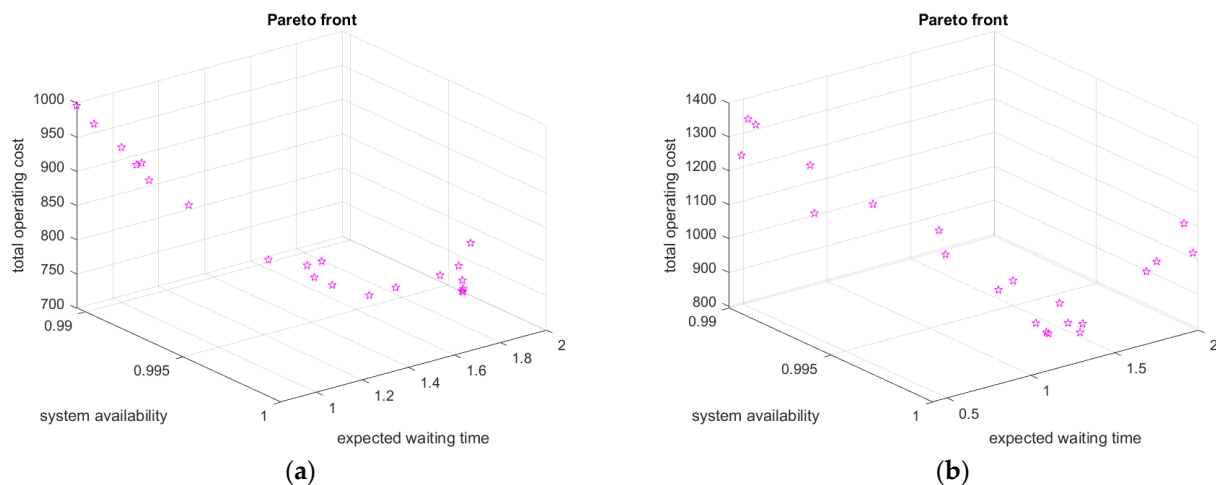


Figure 10. The Pareto-optimal frontiers for (a) $(M, S) = (8, 4)$; (b) $(M, S) = (10, 5)$.

7. Conclusions and Future Research

7.1. Concluding Remarks

This paper has presented a comprehensive operational and financial evaluation of a multi-server retrial machine repair system characterized by warm standby units, proactive orbital search, and a synchronous multi-type vacation mechanism. Prompted by real-world technical resource constraints in high-acuity clinical environments—such as hospital biomedical engineering teams balancing mandatory training with emergency on-call rest duties—we constructed a multi-dimensional, continuous-time Markovian framework. To capture the complex, state-dependent coupling between server status transitions and device backlogs, we modeled the underlying queueing network as a level-dependent (inhomogeneous) quasi-birth-and-death process. To bypass the computational tractability barriers and floating-point errors often encountered in large-scale state spaces, we proposed an efficient recursive matrix method solved via boundary-anchored backward induction. Rigorous model validation under extreme parameter thresholds analytically proved that our model asymptotically collapses into classical single-dimensional reliability templates, verifying the mathematical integrity of our transition structures.

Through extensive parametric sweeps and sensitivity analyses, we unmasked critical non-monotonic, U-shaped congestion trajectories governing the orbit backlog and system risk profiles, revealing that over-accelerating individual service rates without cross-status synchronization can paradoxically amplify transient bottlenecks. To bridge the gap between theoretical steady-state probability matrices and actionable engineering design, we developed a tri-objective non-linear co-optimization framework to balance the conflicting priorities of minimizing total system cost, minimizing equipment flow times, and maximizing system availability. Implemented via the multi-objective genetic algorithm, the resulting 3D Pareto-optimal frontiers exposed a clear, diminishing marginal return horizon and identified a distinct “Reliability-Cost Knee Point.” Crucially, our optimization findings establish that scheduling intelligence can act as a powerful, zero-cost asset substitute for raw human labor.

7.2. Future Research

While this study establishes a robust mathematical and strategic foundation for multi-server retrial maintenance networks, several compelling avenues remain open for future academic investigation:

1. **The Markovian Markov Assumption:** The current formulations rely heavily on birth-and-death stochastic processes if all machine failure, retrial, and repair intervals

follow memoryless exponential distributions. In empirical hospital environments, maintenance tasks often follow deterministic or heavy-tailed patterns. Future research could expand this framework to non-Markovian queueing systems using Phase-Type distributions or the supplementary variable technique.

2. **Homogeneous Technical Capacities:** Our model assumes a unified technician team where all c servers possess identical repair efficiencies (μ, μ_v) . In real-world biomedical engineering teams, staff experience and repair speeds vary significantly. Incorporating heterogeneous server capacities and skill-based routing rules represents a highly promising future research direction.
3. **Synchronous Vacation Constraints:** The multi-type vacation policy evaluated in this study operates under a synchronous discipline, where the entire engineering crew enters administrative or rest states simultaneously. While this mirrors specific small-scale hospital shifts, extending the model to an asynchronous or independent server vacation policy would offer broader applicability for massive, multi-department enterprise server clusters.

Author Contributions: Conceptualization, L.-W.C., T.-H.L., M.-C.C. and F.-M.C.; methodology, T.-H.L. and L.-W.C.; software, T.-H.L. and M.-C.C.; validation, T.-H.L., L.-W.C. and F.-M.C.; writing—original draft preparation, T.-H.L., M.-C.C. and L.-W.C.; writing—review and editing, T.-H.L.; supervision, F.-M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ICU	Intensive care unit
FIFO	First-in-first-out
BME	Biomedical engineering
PSO	Particle swarm optimization
GA	Genetic algorithm
MOGA	Multi-objective genetic algorithm

Appendix A

Each sub-block of the matrix $\mathbf{Q}$ can be provided as follows:

$$\mathbf{B}_0 = \begin{bmatrix} \mathbf{b}_{0,v}^d & \mathbf{b}_{0,v}^u & & & \\ \mathbf{b}_{0,v}^l & \mathbf{b}_{0,0}^d & \mathbf{b}_{0,0}^u & & \\ & \mathbf{b}_{0,1}^l & \mathbf{b}_{0,1}^d & \mathbf{b}_{0,1}^u & \\ & & \ddots & \ddots & \\ & & & \mathbf{b}_{0,c-1}^l & \mathbf{b}_{0,c-1}^d & \mathbf{b}_{0,c-1}^u \\ & & & & \mathbf{b}_{0,c}^l & \mathbf{b}_{0,c}^d \end{bmatrix},$$

$$\mathbf{B}_n = \begin{bmatrix} \mathbf{b}_{n,v}^d & \mathbf{b}_{n,v}^u & & & \\ & \mathbf{b}_{n,0}^d & \mathbf{b}_{n,0}^u & & \\ & & \mathbf{b}_{n,1}^d & \mathbf{b}_{n,1}^u & \\ & & & \ddots & \ddots \\ & & & & \mathbf{b}_{n,c-1}^d & \mathbf{b}_{n,c-1}^u \\ & & & & & \mathbf{b}_{n,c}^d \end{bmatrix}, 1 \leq n \leq K - c,$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_v^d & & & & & \\ & \mathbf{a}_0^d & & & & \\ & \mathbf{a}_1^l & \mathbf{a}_1^d & & & \\ & & \ddots & & & \\ & & & \mathbf{a}_{c-1}^l & \mathbf{a}_{c-1}^d & \\ & & & & \mathbf{a}_c^l & \mathbf{a}_c^d \end{bmatrix},$$

$$\mathbf{C}_n = \begin{bmatrix} \mathbf{d}_{n,v} & & & & & \\ & \mathbf{d}_{n,0} & & & & \\ & & \mathbf{d}_{n,1} & & & \\ & & & \ddots & & \\ & & & & \mathbf{d}_{n,c-1} & \\ & & & & & \mathbf{d}_{n,c} \end{bmatrix}, 0 \leq n \leq K - c - 1.$$

Each block of $\mathbf{A}_n$, $\mathbf{B}_n$, $\mathbf{C}_n$ is provided respectively as follows:

$$\mathbf{b}_{0,v}^d = \begin{bmatrix} -\lambda_0 & & & & & \\ \theta_1 & -\vartheta_0^{0,v} & \lambda_0 & & & \\ & \mu_v & -\vartheta_1^{0,v} & \lambda_1 & & \\ & & \ddots & \ddots & & \\ & & & (c-1)\mu_v & -\vartheta_{c-1}^{0,v} & \lambda_{c-1} \\ & & & & c\mu_v & -\vartheta_c^{0,v} \end{bmatrix},$$

$$\mathbf{b}_{0,0}^d = \begin{bmatrix} -\vartheta_0^{0,1} & & & & & \\ 0 & -\vartheta_1^{0,1} & \lambda_0 & & & \\ & 2\mu & -\vartheta_2^{0,0} & \lambda_1 & & \\ & & \ddots & \ddots & & \\ & & & (c-1)\mu & -\vartheta_{c-1}^{0,0} & \lambda_{c-1} \\ & & & & c\mu & -\vartheta_c^{0,0} \end{bmatrix},$$

$$\mathbf{b}_{0,j}^d = \begin{bmatrix} -\vartheta_0^{0,j} & & & & & \\ \mu & -\vartheta_1^{0,j} & \lambda_0 & & & \\ & 2\mu & -\vartheta_2^{0,j} & \lambda_1 & & \\ & & \ddots & \ddots & & \\ & & & (c-j-1)\mu & -\vartheta_{c-j-1}^{0,j} & \lambda_{c-j-1} \\ & & & & (c-j)\mu & -\vartheta_{c-j}^{0,j} \end{bmatrix}, 1 \leq j \leq c,$$

$$\mathbf{b}_{0,v}^u = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \theta_1 \mathbf{I}_{(c)} \end{bmatrix}, \mathbf{b}_{0,j}^u = \begin{bmatrix} \delta \mathbf{I}_{(c-j+1)} \\ \mathbf{0} \end{bmatrix}, 0 \leq j < c,$$

$$\mathbf{b}_{0,v}^l = \begin{bmatrix} 0 & & & & & \\ & \mu & & & & \\ & & 0 & & & \\ & & & \ddots & & \\ & & & & 0 & 0 \end{bmatrix}, \mathbf{b}_{0,j}^l = [j\eta \mathbf{I}_{(c-j+1)} \quad \mathbf{0}], 1 \leq j \leq c,$$

$$\mathbf{b}_{n,v}^d = \begin{bmatrix} -(\lambda_n + \theta_2) & & & & & \\ & -\vartheta_0^{n,v} & \lambda_n & & & \\ & \mu_v & -\vartheta_1^{n,v} & \lambda_{n+1} & & \\ & & \ddots & \ddots & & \\ & & & (c-1)\mu_v & -\vartheta_{c-1}^{n,v} & \lambda_{n+c-1} \\ & & & & c\mu_v & -\vartheta_c^{n,v} \end{bmatrix},$$

$$\mathbf{b}_{n,j}^d = \begin{bmatrix} -\vartheta_0^{n,j} & \lambda_n & & & & \\ \mu & -\vartheta_1^{n,j} & \lambda_{n+1} & & & \\ & 2\mu & -\vartheta_2^{n,j} & \lambda_{n+2} & & \\ & & & \ddots & & \\ & & & (c-j-1)\mu & -\vartheta_{c-j-1}^{n,j} & \lambda_{n+c-j-1} \\ & & & & (c-j)\mu & -\vartheta_{c-j}^{n,j} \end{bmatrix}, 0 \leq j \leq c,$$

$$\mathbf{b}_{n,v}^u = \begin{bmatrix} [\theta_2, 0, \dots, 0] \\ \theta_1 \mathbf{I}_{(c+1)} \end{bmatrix}, \mathbf{b}_{n,j}^u = \begin{bmatrix} \delta \mathbf{I}_{(c-j+1)} \\ \mathbf{0} \end{bmatrix}, 0 \leq j < c-1,$$

$$\mathbf{a}_v^d = \begin{bmatrix} 0 & 0 & & & & \\ & 0 & \gamma & & & \\ & & 0 & \gamma & & \\ & & & \ddots & \ddots & \\ & & & & 0 & \gamma \\ & & & & & 0 \end{bmatrix},$$

$$\mathbf{a}_j^d = \begin{bmatrix} 0 & 0 & & & & \\ & 0 & \gamma & & & \\ & & 0 & \gamma & & \\ & & & \ddots & \ddots & \\ & & & & 0 & \gamma \\ & & & & & 0 \end{bmatrix}, 0 \leq j \leq c,$$

$$\mathbf{a}_j^l = [\mathbf{0} \quad j\eta \mathbf{I}_{(c-j)}], 1 \leq j \leq c,$$

$$\mathbf{d}_{n,v} = \begin{bmatrix} \lambda_n & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & \lambda_{n+c} \end{bmatrix},$$

$$\mathbf{d}_{n,j} = \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & \lambda_{n+c-j} \end{bmatrix}, 0 \leq j \leq c,$$

$$\vartheta_k^{n,v} = \begin{cases} \lambda_{n+k} + k\mu_v + \theta_1, & 0 \leq k \leq c, n = 0, \\ \lambda_{n+k} + k\mu_v + \theta_1 + \gamma, & 0 \leq k < c, 1 \leq n \leq K - c, \\ \lambda_{n+k} + k\mu_v + \theta_1, & k = c, 1 \leq n \leq K - c, \\ k\mu_v + \theta_1, & k = c, n = K - c, \end{cases}$$

$$\vartheta_k^{n,j} = \begin{cases} \lambda_k + k\mu + j\eta + \delta, & 0 \leq k < c-j, 0 \leq j \leq c, n = 0, \\ \lambda_k + k\mu + j\eta, & k = c-j, 0 \leq j \leq c, n = 0, \\ \lambda_k + k\mu + j\eta + \delta + \gamma, & 0 \leq k < c-j, 0 \leq j \leq c, 1 \leq n \leq K - c, \\ \lambda_k + k\mu + j\eta + \delta, & k = c-j, 0 \leq j \leq c, 1 \leq n \leq K - c, \\ k\mu + j\eta, & k = c-j, 0 \leq j \leq c, n = K - c. \end{cases}$$

References

1. D'Apice, C.; Dudin, A.; Dudin, S.; Manzo, R. Analysis of a multi-server retrial queue with a varying finite number of sources. *AIMS Math.* **2024**, *9*, 33365–33385. <https://doi.org/10.3934/math.20241592>.
2. Dudin, A.; Dudina, O.; Samouylov, K. Analysis of BMAP/PH/N-type queueing system with flexible retrials admission control. *Mathematics* **2025**, *13*, 1434. <https://doi.org/10.3390/math13091434>.

3. Kuo, C.C.; Sheu, S.H.; Ke, J.C.; Zhang, Z.G. Reliability-based measures for a retrial system with mixed standby components. *Appl. Math. Model.* **2014**, *38*, 4640–4651. <https://doi.org/10.1016/j.apm.2014.03.005>.
4. Yang, D.Y.; Tsao, C.L. Reliability and availability analysis of standby systems with working vacations and retrial of failed components. *Reliab. Eng. Syst. Saf.* **2019**, *182*, 46–55. <https://doi.org/10.1016/j.res.2018.09.020>.
5. Jain, M.; Sanga, S.S. Fuzzy cost optimization and admission control for machine interference problem with general retrial. *J. Test. Eval.* **2020**, *48*, 4488–4502. <https://doi.org/10.1520/JTE20180882>.
6. Liu, S.; Hu, L.; Liu, Z.; Wang, Y. Reliability of a retrial system with mixed standby components and Bernoulli vacation. *Qual. Technol. Quant. Manag.* **2021**, *18*, 248–265. <https://doi.org/10.1080/16843703.2020.1853320>.
7. Gao, S.; Wang, J. Reliability and availability analysis of a retrial system with mixed standbys and an unreliable repair facility. *Reliab. Eng. Syst. Saf.* **2021**, *205*, 107240. <https://doi.org/10.1016/j.res.2020.107240>.
8. Hu, L.; Liu, S.; Peng, R.; Liu, Z. Reliability and sensitivity analysis of a repairable k-out-of-n: G system with two failure modes and retrial feature. *Commun. Stat. Theory Methods* **2022**, *51*, 3043–3064. <https://doi.org/10.1080/03610926.2020.1788083>.
9. Sanga, S.S.; Charan, G.S. Fuzzy modeling and cost optimization for machine repair problem with retrial under admission control F-policy and feedback. *Math. Comput. Simul.* **2023**, *211*, 214–240. <https://doi.org/10.1016/j.matcom.2023.03.036>.
10. Gao, S. Availability and reliability analysis of a retrial system with warm standbys and second optional repair service. *Commun. Stat. Theory Methods* **2023**, *52*, 1039–1057. <https://doi.org/10.1080/03610926.2021.1922702>.
11. Kumar, K.; Jain, M.; Shekhar, C. Machine repair system with threshold recovery policy, unreliable servers and phase repairs. *Qual. Technol. Quant. Manag.* **2024**, *21*, 587–610. <https://doi.org/10.1080/16843703.2023.2232639>.
12. Wu, C.H.; Yang, D.Y.; Ko, M.H. Performance sensitivity analysis for machine repair problem with two failure modes and working vacation. *Int. J. Reliab. Qual. Saf. Eng.* **2024**, *31*, 2350040. <https://doi.org/10.1142/S0218539323500407>.
13. Shao, Q.; Hu, L.; Xu, F. Reliability and optimization for k-out-of-n: G mixed standby retrial system with dependency and J-vacation. *Methodol. Comput. Appl. Probab.* **2024**, *26*, 8. <https://doi.org/10.1007/s11009-024-10078-x>.
14. Chen, W.L.; Koh, T.; Lin, J.H. Reliability analysis and sensitivity analysis of retrial machine repair problems with multiple vacations and the recovery policy. *Int. J. Reliab. Qual. Saf. Eng.* **2025**, *32*, 2450045. <https://doi.org/10.1142/S0218539324500451>.
15. Mathavavisakan, N.M.; Indhira, K.; Ammar, S.I. Cost optimization and analysis of retrial queues with hybrid vacation policies and fault correction: A metaheuristic approach. *RAIRO Oper. Res.* **2025**, *59*, 2683–2720. <https://doi.org/10.1051/ro/2025090>.
16. Shao, Q.; Hu, L.; Xu, F.; Li, J.; Zhong, Y. Redundancy optimization and reliability evaluation of (k1, k2)-out-of-n: G retrial system with redundant dependency and working vacation. *Qual. Technol. Quant. Manag.* **2026**, *23*, 326–354. <https://doi.org/10.1080/16843703.2025.2502902>.
17. Wu, C.H.; Wang, K.H. Multi-objective optimization of retrial warm-standby machine repair system with two failure modes. *Qual. Technol. Quant. Manag.* **2026**, *23*, 303–325. <https://doi.org/10.1080/16843703.2025.2499801>.
18. Neuts, M.F.; Ramalhoto, M.F. A service model in which the server is required to search for customers. *J. Appl. Probab.* **1984**, *21*, 157–166. <https://doi.org/10.2307/3213673>.
19. Ahammed, S.N.; Rajadurai, P.; Indhira, K.; Chandrasekaran, V.M. A repairable retrial queueing system with two types of service, modified vacations and orbit search. *Int. J. Appl. Eng. Res.* **2014**, *9*, 4337–4357.
20. Li, J.; Li, T. An M/M/1 retrial queue with working vacation, orbit search and balking. *Eng. Lett.* **2019**, *27*, 97–102.
21. Nila, M.; Sumitha, D. M/G/1 retrial queueing model with pre-emptive resume priority, multi optional second phase, feedback and orbital search. *Int. J. Sci. Technol. Res.* **2020**, *9*, 521–524.
22. Gao, S.; Wang, J. Stochastic analysis of a preemptive retrial queue with orbital search and multiple vacations. *RAIRO Oper. Res.* **2020**, *54*, 231–249. <https://doi.org/10.1051/ro/2018117>.
23. Kumar, B.K.; Sankar, R.; Krishnan, R.N.; Rukmani, R. Multiserver call center retrial queue under Bernoulli vacation schedule with two-way communication and orbital search. *Telecommun. Syst.* **2023**, *84*, 23–51. <https://doi.org/10.1007/s11235-023-01025-1>.
24. Chang, F.M.; Liu, T.H.; Ke, J.C. On an unreliable-server retrial queue with customer feedback and impatience. *Appl. Math. Model.* **2018**, *55*, 171–182. <https://doi.org/10.1016/j.apm.2017.10.025>.
25. Xu, F.; Tian, R.; Hu, L.; Shao, Q. Bi-objective optimization and reliability analysis in unreliable M/G/1 retrial queues with rational customers, reserved idle times and setup times. *Qual. Technol. Quant. Manag.* **2025**, *22*, 385–417. <https://doi.org/10.1080/16843703.2024.2328991>.
26. Kumar, K.; Garg, S.; Sharma, S. Cost optimization for a retrial repairable queueing healthcare system with working vacation and unreliability using the genetic algorithm and particle swarm optimization. *Oper. Res. Data Anal. Logist.* **2025**, *45*, 200473. <https://doi.org/10.1016/j.ordal.2025.200473>.

27. Ammar, S.I.; Rajadurai, P. Performance analysis of preemptive priority retrial queueing system with disaster under working breakdown services. *Symmetry* **2019**, *11*, 419. <https://doi.org/10.3390/sym11030419>.
28. Saravanan, V.; Poongothai, V.; Godhandaraman, P. Performance analysis of a retrial queueing system with optional service, unreliable server, balking and feedback. *Int. J. Math. Eng. Manag. Sci.* **2023**, *8*, 769–786. <https://doi.org/10.33889/IJMEMS.2023.8.4.044>.
29. Shanmugam, B.; Saravananarajan, M.C. Unreliable retrial queueing system with working vacation. *AIMS Math.* **2023**, *8*, 24196–24224. <https://doi.org/10.3934/math.20231234>.
30. Sundararaman, M.; Narasimhan, D.; Rajadurai, P. Performance Analysis of Two Different Types of Waiting Queues with Working Vacations, *J. Math.* **2024**, *2024*, 5829171. <https://doi.org/10.1155/2024/5829171>.
31. Liu, T.H.; Hsu, H.Y.; Chang, F.M. Multi-server tow-way communication retrial queue subject to disaster and synchronous working vacation. *Algorithms* **2025**, *18*, 24. <https://doi.org/10.3390/a18010024>.
32. Wang, K.H.; Wu, C.H.; Yen, T.C. Comparative cost-benefit analysis of four retrial systems with preventive maintenance and unreliable service station. *Reliab. Eng. Syst. Saf.* **2022**, *221*, 108342. <https://doi.org/10.1016/j.res.2022.108342>.
33. Wang, K.H.; Wu, C.H.; Yen, T.C. Reliability analysis of redundant retrial machining system subject to standby switching failure, *Qual. Technol. Quant. Manag.* **2023**, *20*, 561–576. <https://doi.org/10.1080/16843703.2022.2132450>.
34. Liu, Z.; Wu, J. Reliability analysis and cost optimization of a redundant retrial machine repair system. *Reliab. Eng. Syst. Saf.* **2027**, *277*, 112956. <https://doi.org/10.1016/j.res.2026.112956>.
35. Sivazlian, B.D.; Wang, K.H. Economic analysis of the M/M/R machine repair problem with warm standbys. *Microelectron. Reliab.* **1989**, *29*, 25–35. [https://doi.org/10.1016/0026-2714\(89\)90007-3](https://doi.org/10.1016/0026-2714(89)90007-3).
36. Kumar, A.; Jain, M. Cost optimization of an unreliable server queue with two stage service process under hybrid vacation policy. *Math. Comput. Simul.* **2023**, *204*, 259–281. <https://doi.org/10.1016/j.matcom.2022.08.007>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.