SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/

e-ISSN: 2636-8129
Publisher: Sakarya University

Vol. 9, No.3, 650-673, 2026
DOI:10.35377/saucis...1727731

SAKARYA UNIVERSITY

RESEARCH ARTICLE

Optimization of Emergency Healthcare System with Priority Based Queuing Model and Dynamic Programming

Abdela Atisso Mohammed,[1*], **Natesan Thillaigovindan,**[1],**Endalkachew Teshome Ayele** [1]

[1]Department of Mathematics, Arba Minch University, Ethiopia

*Corresponding author:
abdela_atisso@yahoo.com

Article History:
Received: 26 June 2025
Revised: 15 Jaunary 2026
Accepted: 23 Jaunary 2026
Published Online: 18 June 2026

Abstract
This study introduces a priority-based multi-class, multi-server accumulating priority queuing system designed to balance waiting times and resource utilization under both normal and high-demand conditions. Dynamic adjustment of priority rates ensures minimal delays for high-priority patients during peak periods while maintaining fairness across patient groups. System performance is evaluated using metrics such as queue length, priority accumulation, and a fairness index that capture complex, nonlinear interactions between resource allocation strategies and patient flow. Integration of these performance metrics strengthens the model's ability to respond to real-world variability and ensures robust performance under unpredictable demand. Advanced queuing frameworks, including Markov-modulated and non-Markovian models, further reveal trade-offs between fairness and efficiency, providing additional insights into adaptive healthcare system design. Class-specific, metric-focused simulations validate system reliability and effectiveness. In addition, dynamic programming is employed to optimize resource allocation by balancing urgency and waiting time, thereby enhancing cost efficiency and fairness. Visualizations such as resource utilization plots and patient distribution graphs support informed and effective management decisions.

Keywords: Dynamic programming, Healthcare, Heterogeneous waiting time distributions, Multi-class Multi-server Accumulating Priority Queue, Non-linear priority, Simulation

1. Introduction

Efficient operation of emergency departments (EDs) is critical to overall hospital performance, particularly in resource-constrained healthcare systems where demand uncertainty, congestion, and limited staffing are persistent challenges. These challenges have motivated extensive research on queuing-based models for healthcare systems, including priority queues, multi-server systems, and scheduling-based control policies [1–3]. This exhibits the difference between classical queuing theoretical assumptions and the stochastic and dynamic nature of hospital operations [4]. Classical queuing models such as First-Come–First-Served (FCFS), fixed-priority queues, and standard APQs provide important viewpoints but remain limited in their ability to simultaneously address urgency, fairness, and time-varying system conditions. In particular, fixed-priority rules can cause excessive delays for low-priority patients, whereas traditional APQs are often analyzed under restrictive assumptions that neglect demand variability, server heterogeneity, and adaptive control mechanisms. In [5, 6], it is observed that long waiting times are a serious concern in many public health systems. According to [7], prioritizing patients is a system used to control access to Healthcare. Thus, prioritizing the decision-making process is supported by patient prioritization tools (PPT), which also help to ensure fairness and openness. They have also asserted that tools for patient prioritization may differ depending on the specific application. For example, the classical priority queuing discipline commonly used by physicians and decision makers in health care systems is the Canadian Triage and Acuity Scale (CTAS) [8]. This study addresses these limitations by developing a unified model and optimization framework for emergency healthcare systems based on a Multi-class, Multi-server Accumulating Priority Queue (MMAPQ) integrated with stochastic control techniques. While the individual components of this

Cite as: AA Mohammed, N Thillaigovindan, and ET Ayele, "Optimization of emergency healthcare system with priority based queuing model and dynamic programming" , *Sakarya University Journal of Computer and Information Sciences* 9 (3) 2026, 650-673. doi: 10.35377/saucis...1727731

framework, such as APQs, Markov-modulated processes, fairness metrics, and dynamic programming, are well established in the literature, their joint integration and systematic evaluation in a healthcare context remain underexplored. The scholarly contribution of this study is threefold. First, we develop an extended MMAPQ framework that jointly accounts for Markov-modulated arrivals and non-Markovian service mechanisms, enabling the realistic modeling of time-varying demand, stochastic service behavior, and heterogeneous patient classes within a single analytical framework. Second, the study introduces a fairness-aware optimization perspective by embedding a Gini-like heterogeneity index directly into the queuing and control framework, allowing explicit and quantitative assessment of efficiency equity trade-offs that are largely absent in classical and fixed-priority models. Third, we formulate and implement a dynamic programming–based adaptive control policy that dynamically adjusts priority accumulation rates in response to congestion and system state. Unlike existing studies that focus primarily on analytical properties, the proposed approach is rigorously validated through simulation and systematically benchmarked against FCFS and fixed-priority queues, demonstrating measurable improvements in waiting times, robustness, and fairness under high-load ED conditions.

2. Related Literature and Preliminaries

Dynamic programming is a fundamental methodology for analyzing and optimizing queuing systems, including both single-class and multi-class multi-server models. In single-class queues, it is commonly used to derive optimal control policies for resource allocation and system operation with the objective of minimizing average waiting times or maximizing throughput. Its applicability naturally extends to multi-class multi-server systems, where heterogeneous customer classes interact with multiple servers having distinct service characteristics. In such settings, optimal system performance requires careful consideration of class–server interactions [9]. In queuing applications, dynamic programming typically involves defining a state space that captures the system status, specifying decision variables that represent admissible actions, and formulating recursive equations to compute optimal policies. This framework provides both analytical insight and practical control strategies for improving queue performance [10]. Stochastic dynamic programming is commonly formulated as a Markov decision process (MDP), or a Markov decision chain in discrete time, characterized by states, actions, costs, transition probabilities, state equations, and an optimization criterion. A comprehensive treatment of stochastic dynamic programming for time-varying single-server queues is provided in [11]. Markov-modulated single-server queues driven by discrete-time Markov chains were studied in [12], where joint distributions of waiting and idle times were derived, with the Markov-modulated M/M/1 system analyzed as a special case. A GI/M/m model with parallel servers subject to random service interruptions was examined in [13], where limiting distributions were obtained using embedded Markov chains. Methods and software for evaluating queuing approximations based on event-driven state transitions were proposed in [10]. Markov-modulated stochastic recursive equations and their stationary and transient properties were investigated in [14]. Customer impatience in single-server queues was modeled in [15] using an age-based Markov process. Their relevance stems from their impact on response times and system efficiency. Optimal scheduling policies for queues with geometric service times and preemptive disciplines were analyzed in [16], while strong conservation laws for multi-class queuing systems were established in [17], greatly simplifying optimal control analysis. Polling systems with routing and service policies were studied in [18], and preemptive server assignment for two-queue systems was examined, where threshold-type policies were shown to be effective. In [19, 20], waiting-time distributions were derived for multi-server APQs with heterogeneous servers, and a measure of server heterogeneity based on Gini-like coefficients was introduced. Dispatching policies for multi-server systems with job-dependent resource requirements were studied in [21], leading to optimal scheduling rules and tail performance results. Priority-based dispatching in information transmission systems and emergency services was discussed in [22].

Definition 2.1 [23]. Consider a two-class accumulating priority queue (APQ). The maximum priority process is defined by

$$M = \{(M_1(t), M_2(t)),\ t \geq 0\},$$

where $M_k(t)$ denotes the maximum accumulated priority of class k customers at time t, for $k = 1, 2$.

1. **Empty queue case:**
   If the system is empty at time t, that is,

   $$t \in (D_{n(x)}, \tau_{n(x+1)})$$

   for some x, then

   $$M_1(t) = M_2(t) = 0.$$

2. **At departure epochs:**
   For the sequence of departure times

   $$\{D_{n(x)},\ x = 0, 1, 2, \ldots\},$$

   the maximum priority levels are defined as

   $$M_1(D_{n(x)}) = \max_{n \notin \{n(k):1 \leq k \leq x\}} V_n(D_{n(x)}), \tag{2.1}$$

   and

   $$M_2(D_{n(x)}) = \min\left\{M_1(D_{n(x)}),\ M_2(C_{n(x)}) + b_2 X_{n(x)}\right\}. \tag{2.2}$$

3. **During service intervals:**
   For

   $$t \in (C_{n(x)}, D_{n(x)})$$

with

$$P_x(t) > 0,$$

that is, when customers are present in the system, the priority process evolves according to

$$M_k(t) = M_k(C_{n(x)}) + b_k\big(t - C_{n(x)}\big), \qquad k = 1,2. \tag{2.3}$$

The above definition can also be illustrated using the flowchart presented in Figure 1.

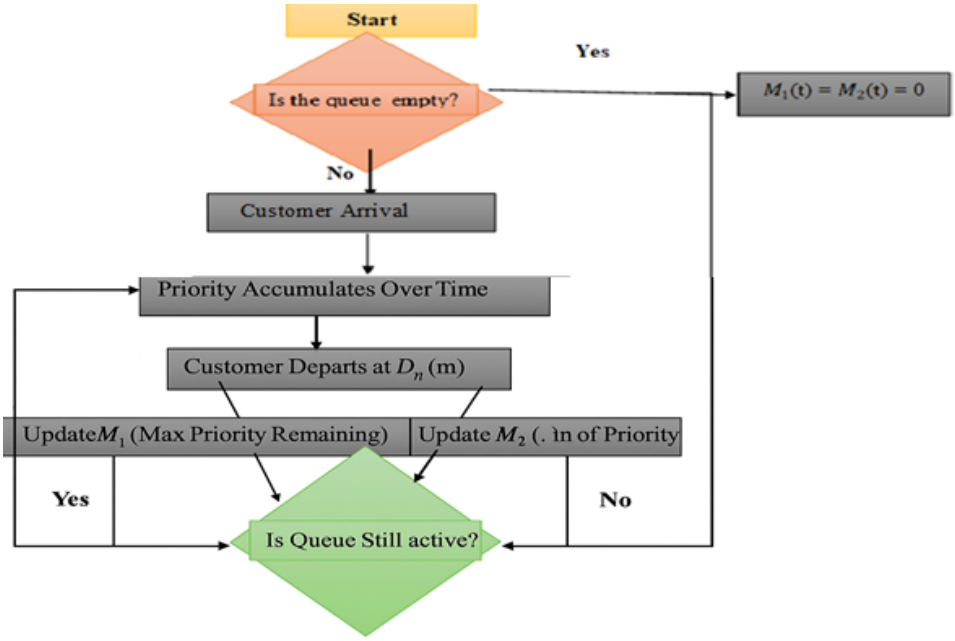

Figure 1. Flow chart

Theorem 2.1 [1]. Let $t \in [0,\infty)$. Given the maximum priority process

$$M(t) = \max\{M_1(t), M_2(t)\}$$

at time t, the accumulated priorities

$$\{P_k^i(t), k = 1,2,\ldots\}$$

of customers from class i still waiting in the queue, for $i = 1,2$, are distributed as independent Poisson processes with rates

$$\frac{\lambda_i}{b_i}$$

over the intervals

$$[0, M_i(t)).$$

For all customers present in the queue, the accumulated priorities

$$\{P_k(t), k = 1,2,\ldots\}$$

are distributed as a Poisson process with a piecewise constant rate defined as follows:

- zero on the interval

  $$[M_1(t), \infty),$$

- rate

  $$\frac{\lambda_1}{b_1}$$

  on the interval

  $$[M_2(t), M_1(t)),$$

- and rate

$$\frac{\lambda_1}{b_1}+\frac{\lambda_2}{b_2}$$

on the interval

$$[0, M_2(t)).$$

A waiting customer with priority

$$P \in [0, M_2(t))$$

belongs to class 1 with probability

$$\frac{\lambda_1 b_2}{\lambda_2 b_1 + \lambda_1 b_2},$$

independently of the class of other customers in the queue.

The rate at which these priorities accumulate is determined by the arrival rate λ_i divided by the priority accumulation rate b_i for that class.

Reference [19] defined a vector of Gini-like coefficients, denoted by

$$G = \{G_i\}_{i=1}^{C},$$

to evaluate the level of heterogeneity in the multi-server system. The ith index is defined by

$$G_i = \frac{\mu_1 - \mu_i}{\mu_1 + \mu_i}, \qquad i = 1, 2, \ldots, C.$$

The total cost function is defined as

$$F(C, r, G) = \frac{\pi(\mu, c; r)}{\mu_a} \cdot \frac{\rho}{1-\rho},$$

where

$$\pi(\mu, c; r) = \sum_{i=1}^{C} \mu_i^{r-1},$$

in order to determine the optimal level of heterogeneity G that minimizes the cost and the optimal range of G for which the r-dispatch policies have lower cost than the homogeneous system.

Corollary 2.1 [24, 25]. In any multi-server queue with Poisson arrivals and heterogeneous exponential service distributions $(M/M_i/c)$ under any non-preemptive work-conserving queue discipline, for a given r-dispatch policy,

$$F(C, r, G) = \sum_{k=1}^{K} \rho_k m_k(C, r, G) = \frac{\pi(\mu, c; r)}{\mu_a} \cdot \frac{\rho}{1-\rho}, \qquad \rho < 1,$$

where μ_a denotes the aggregate service rate.

The analytic operations for the case of three or more servers are generally too complicated to establish the influence of specific parameters. This observation holds under the slowest-server-first (SSF) and randomly-chosen-server (RCS) dispatch policies. However, there exists a range of G values in the two-server case for which the cost function decreases under the fastest-server-first (FSF) dispatch policy [12].

Lemma 2.1 [24]. In a two-server queue operating under any work-conserving queue discipline:

- For $r \leq 0$ (i.e., $p_1 \leq p_2$), the minimum value of the cost function $F(2, r, G)$ occurs in the homogeneous case.
- For $r > 0$ (i.e., $p_1 \geq p_2$), the optimal value of the decision variable G that minimizes the cost function

$$F(2, r, G) = \frac{2\rho^3\big(2\rho + 1 - G(p_1 - p_2)\big)}{(1-\rho)\left[2(G^2+1)\rho^2 + (3 - G^2 + 2G(p_1 - p_2))\rho + 1 - G^2\right]}$$

is given by

$$G^* = \frac{2\rho + 1 - \sqrt{(2\rho+1)^2 - (p_1 - p_2)^2}}{p_1 - p_2}. \tag{2.4}$$

When r is sufficiently large, an optimal range of G exists for which the function $F(2,r,G)$ improves upon the homogeneous case. The optimal range is

$$G \in \left(0, \frac{p_1 - p_2}{1+2\rho}\right).$$

Corollary 2.2 [24]. Consider a two-server queue operating under a work-conserving queue discipline and define

$$r^* = \frac{\ln\big(\mu_1 + \rho(\mu_1 - \mu_2)\big) - \ln\big(\mu_2 - \rho(\mu_1 - \mu_2)\big)}{\ln(\mu_1) - \ln(\mu_2)}. \tag{2.5}$$

Then an improvement over the homogeneous system with the same aggregate service rate can be achieved if and only if

$$r > r^*.$$

2.1. Average waiting times in APQs

Each curve in Figure 2 represents a different growth rate . Higher means faster urgency build-up. For =1.0 urgency reaches maximum within the time t=10, for =0.1 urgency builds very slowly and it takes 30–40 plus time to approach W$_{m}ax.Thecurvesasymptoticallyapproachthesamevalueat W_{m}ax = 10.Table1belowisobtainedbystepbystepevaluationofnumericalexampleusingEulermethod(t=1).Lettheinitialpriorityfunctionforclass2patientbeP_2(0)$ $2andtimestept = 0,1,2,...,10.TheEulermethodisusedtodiscretizetheintegralP_2(t) = P_2(0) +_0^t g_2(W_2())d.TheEulerapproachforthegivenintegralisP_2(t_(n-$ $1)) = P_2(t_n) + tg_2(W_2(t_n)).Asimilartechniquecanbeusedforclass1andclass3patients.$

For the $M/M/1$ linear accumulating priority queue (APQ), [26] derived a set of recursive formulas for the average waiting times of different customer classes. Assume that customers from class k require service times that are exponentially distributed with mean.

$$\frac{1}{\mu_k},$$

for $k = 1,2,\ldots,K$.

The average waiting time for class-k customers, denoted by m_k, is obtained recursively for

$$k = K, K-1, \ldots, 1,$$

as follows:

$$m_K = \frac{\dfrac{w_0}{1-\rho}}{1 - \displaystyle\sum_{j=1}^{K-1} \rho_j \left(1 - \frac{b_K}{b_j}\right)},$$

and

$$m_k = \frac{\dfrac{w_0}{1-\rho} - \displaystyle\sum_{j=k+1}^{K} \rho_j m_j \left(1 - \frac{b_j}{b_k}\right)}{1 - \displaystyle\sum_{j=1}^{k-1} \rho_j \left(1 - \frac{b_k}{b_j}\right)}, \qquad k = 1,2,\ldots,K-1,$$

where

$$w_0 = \sum_{k=1}^{K} \frac{\rho_k}{\mu_k},$$

$$\rho_k = \frac{\lambda_k}{\mu_k},$$

and

$$\rho = \sum_{k=1}^{K} \rho_k < 1.$$

The above formulation is also applicable to the $M/G/1$ case, where the term w_0 is given by

$$w_0 = \sum_{k=1}^{K} \frac{\lambda_k}{\mu_k^2} = \sum_{k=1}^{K} \frac{\lambda_k \overline{x_k^2}}{2},$$

where

$$\overline{x_k^2}$$

denotes the second moment of the service time distribution for customers from class k.

2.2. The M/M/1 power-law APQ of order

As stated in [27], the average waiting time for each class of customers in a single-server APQ with Poisson arrivals, exponential service times, and a set of power law accumulation functions was analytically derived to find the optimal time. The priority accumulation functions for the power-law APQ of order r are defined in terms of a sequence $\{b_k^{(r)}\}, k = 1, \ldots, K$ of positive constants such that

$$b_1^{(r)} \geq b_2^{(r)} \geq \cdots \geq b_K^{(r)} \geq 0,$$

and with a functional form

$$b_k^{(r)} t^r$$

for all k. They established that if one selects the constants such that

$$\left(\frac{b_{k+1}^{(r)}}{b_k^{(r)}} \right)^{1/r} = \left(\frac{b_{k+1}^{(r')}}{b_k^{(r')}} \right)^{1/r'}$$

for $k = 1, 2, \ldots, K$, then the expected waiting times of all customer classes in the corresponding power-law APQs of orders r and r' would be identical. From this, using the results in [28] for the first-order systems, they obtained the expected waiting times for different classes of customers in a power-law APQ of order r.

2.3. Waiting time distributions in APQs

In the context of an $M/G/1$ linear APQ, [20] determined waiting time distributions for each class of customers in a single-server linear APQ with Poisson arrivals and general service time distributions. A key element in their derivation was the stochastic process named the maximum priority process,

$$M = \max\{M_i(t); t \geq 0,\ i = 1, \ldots, K\},$$

which is the least upper bound of the accumulated priority $M_i(t)$ for each priority class at a given instant.

They began with the accumulating priority queue in the two-class case with

$$M = \max\{M_1(t), M_2(t); t \geq 0\},$$

where $M_i(t)$, $i = 1, 2$, is an upper bound on the value of the maximum accumulated priority for the class-i customer, with

$$M_1(t) \geq M_2(t)$$

for all t.

A class-1 customer in the queue with accumulated priority at time t that lies in the interval

$$(M_2(t), M_1(t)]$$

is referred to as "accredited relative to class-2", or simply "accredited", since it is certain that there are no class-2 customers in the system with a higher priority. Those customers with priority in the interval

$$[0, M_2(t))$$

are referred to as "non-accredited".

Corollary 2.4 [19]. During a busy period, the times at which customers from class $i \leq k$ are accredited at level k follow a Poisson process with rate

$$\frac{\lambda_i(b_i - b_{k+1})}{b_i}.$$

Consequently, the times at which customers from any of the classes $i \leq k$ receive accreditation at level k are collectively distributed as a Poisson process with rate.

$$A_k = \sum_{i=1}^{k} \frac{\lambda_i(b_i - b_{k+1})}{b_i}.$$

A customer from class $j \leq k$ is considered to be served at accreditation level k if their priority lies within the interval.

$$[M_{k+1}(t), M_k(t))$$

at the time they begin service.

An accreditation interval at level k begins either at the start of a busy period or when a customer is served at an accreditation level $l_1 > k$, and it ends either at the end of the busy period or when another customer is served at an accreditation level $l_2 > k$.

2.4. The M/M/c linear APQ

A multi-class multi-server linear APQ with Poisson arrivals and a common exponential service distribution for all classes of customers with equal service rates, i.e.,

$$\mu_1 = \mu_2 = \cdots = \mu_K = \mu,$$

is analyzed in [24]. They also commented on how to choose feasible accumulation rates to satisfy KPI targets in [29]. As observed in [4], dynamic adjustment of priority accumulation rates is an important mechanism for efficiency, especially during periods of high demand, wherein it minimizes excessive delays of high-priority classes as well.

3. Methodology

3.1. Model Description, Design Assumptions and Optimization in APQ Systems

In this section, we present methods that are designed to support the mathematical models developed to address the objectives of this study. The mathematical model of a Multi-Class Multi-Server Accumulating Priority Queuing (MMAPQ) system offers a significant contribution to the fields of operations research and healthcare system optimization.

The following parameters are used throughout this study:

- C: Total number of servers (e.g., doctors, nurses)
- N: Number of patient classes ($i = 1, 2, \ldots, N$) based on urgency or priority
- $\lambda_i(t)$: Arrival rate of class-i patients at time t
- μ_i: Service rate for class-i patients
- $P_i(t)$: Priority weight of class-i patients at time t
- $W_i(t)$: Waiting time of class-i patients at time t
- $Q_i(t)$: Queue length of class-i patients at time t
- $S(t)$: Total number of patients in the system at time t

Table 1(see next to Bibliography Section) presents the simulation setting parameter descriptions along with their corresponding values, ranges, and units used in this study.

The priority for class i patients over time is defined as:

$$P_i(t) = P_i(0) + \int_0^t g_i(W_i(\tau))\, d\tau,$$

where $g_i(W_i(\tau))$ is a monotonic function that increases with waiting time $W_i(\tau)$. This ensures that patients waiting longer gain higher priority over time. $P_i(0)$ is the initial priority value for class i, and $g_i(W_i(\tau))$ determines the rate of change of the priority based on the waiting time $W_i(\tau)$, while τ represents the time variable.

System performance is evaluated using average and weighted waiting times, server utilization, resource efficiency, and a Gini-like index measuring arrival heterogeneity across classes. These metrics provide a detailed assessment of both efficiency and equity.

Definition 3.1. The convergence of the accumulating priority function refers to the property whereby the priority function $P_i(t)$ for a given class i stabilizes to a finite value over time under certain conditions. Mathematically, the priority function is expressed as

$$P_i(t) = P_i(0) + \int_0^t g_i(W_i(\tau))\, d\tau.$$

Table 1. Baseline simulation parameters for MMAPQ system

Parameter	Description	Value / Range	Units
$\lambda_i(t)$	Base arrival rate of class i patients	[5, 3, 2]	patients/hour
μ_i	Service rate for class i	[2, 4, 3]	patients/hour
C	Number of servers	3	servers
N	Number of patient classes	3	-
T	Simulation horizon	10	hours
Δt	Simulation time step	0.01	hours
α_i	Priority accumulation coefficient	[0.1, 0.05, 0.02]	per hour
Priority_weights	Initial priority for each class	[1, 0.5, 0.2]	-
Modulation_rate	Markov transition rate between high/low states	0.5	per hour
High_factor	Multiplier for high arrival state	1.5	-
Low_factor	Multiplier for low arrival state	0.5	-
Erlang_k	Shape parameter for Erlang service times	[2, 2, 2]	-
G_i	Gini-like fairness index	[0,1]	-

The function $P_i(t)$ converges when a suitable growth function is used. A growth function is more suitable for systems that require urgent service, have long waiting times, or involve critical cases; however, a bounded (or saturating) function is preferable to avoid explosive priority inflation and to maintain fair resource allocation.

$$g_i(W_i(\tau)) = \begin{cases} \alpha_i e^{\beta_i w_i}, & \text{for exponential growth} \\ \dfrac{\alpha_i}{1+\beta_i w_i}, & \text{for bounded (or saturating)} \end{cases} \tag{3.1}$$

Equation (3.1) defines two possible forms of the accumulation function $g_i(W_i(\tau))$: an exponential growth model suitable for high-urgency or critical-service cases, and a bounded (saturating) model that prevents uncontrolled growth in priority. These formulations are consistent with accumulating priority queue designs in healthcare and service systems [21, 24], where bounded growth functions are preferred to ensure fairness in multi-class environments.

Theorem 3.1 (Convergence of Accumulating Priority Function). The priority function $P_i(t)$ for class i converges to a steady-state value if $g_i(W_i(\tau))$ is continuous and bounded.

Proof. Assume that the function g_i is continuous and bounded over the domain $t > 0$. Since g_i is bounded, the growth of $P_i(t)$ is limited. Consider the integral

$$\int_0^t g_i(W_i(\tau))\, d\tau.$$

As $t \to \infty$, the boundedness of g_i ensures that the integral remains finite. If $0 < g_i(W_i(\tau)) < M$ for all τ, then $P_i(t)$ is increasing. If $-M < g_i(W_i(\tau)) < 0$, then $P_i(t)$ is decreasing, implying $-M \le g_i(W_i(\tau)) \le M$. Thus the priority function $P_i(t)$ converges to a steady-state value

$$\lim_{t\to\infty} P_i(t).$$

The steady-state value is unique since the integral depends only on the initial condition $P_i(0)$ and the trajectory of $W_i(\tau)$ determined by the system dynamics. This completes the proof.

Lemma 3.1 (Effectiveness of Accumulating Priority). The accumulating priority mechanism ensures that for any two classes i and j with $i < j$, if $W_i(t) > W_j(t)$ initially, there exists a $t_0 > 0$ such that $P_i(t) > P_j(t)$ for all $t > t_0$.

Proof. Assume $W_i(0) > W_j(0)$. Since $g_i(W_i(\tau))$ is increasing in $W_i(\tau)$, it follows that $g_i(W_i(\tau)) > g_j(W_j(\tau))$. Now consider

$$P_i(t) - P_j(t) = P_i(0) - P_j(0) + \int_0^t \left[g_i(W_i(\tau)) - g_j(W_j(\tau))\right] d\tau. \tag{3.2}$$

Since the integrand is positive, $\Delta P(t) = P_i(t) - P_j(t)$ increases over time. Hence, there exists $t_0 > 0$ such that $\Delta P(t) > 0$ for all $t > t_0$. Monotonicity implies persistence of ordering, completing the proof.

Definition 3.2a. The stability of a multi-class multi-server queuing system refers to the condition under which the system can handle incoming workloads without unbounded queue growth over time.

Definition 3.2b. For a multi-class multi-server system with arrival rates $\lambda_i(t)$, service rates μ_i, and priority function $P_i(t)$, the system is stable if

$$\sum_{i=1}^{N} \lambda_i(t) < C \cdot \mu,$$

where $\mu = \min_i \mu_i$.

Theorem 3.2. A multi-class multi-server queuing system is stable if

$$\sum_{i=1}^{N} \lambda_i(t) < C \cdot \mu.$$

Proof. For stability, the traffic intensity satisfies $\rho = \lambda/\mu < 1$, where $\lambda = \sum_{i=1}^{N} \lambda_i(t)$ and $\mu = C \cdot \mu_{\min}$. Hence stability requires

$$\sum_{i=1}^{N} \lambda_i(t) < C \cdot \min_i \mu_i.$$

Thus, if the arrival rate exceeds the service capacity, the system becomes unstable; otherwise, it remains stable.

Lemma 3.2 (Boundedness of Queue Length). If the system is stable, then

$$\lim_{t\to\infty} Q_i(t) \leq \frac{\lambda_i}{\mu_i(1-\rho)}.$$

Proof. Consider the queue length $Q_i(t)$ for class i, defined according to a Volterra-type integral equation as

$$Q_i(t) = Q_i(0) + \int_0^t (\lambda_i(\tau) - \mu_i h_i(Q_i(\tau), P_i(\tau), S(\tau)))\, d\tau \tag{3.3}$$

where

$$h_i(Q_i(\tau), P_i(\tau), S(\tau)) \in [0,1]$$

adjusts the service rate based on priority and system state.

The effective service rate for the class i patients is given by

$$\mu_i h_i(Q_i, P_i, S),$$

where h_i represents the fraction of time servers are available for class i.

By stability, we know

$$\lambda_i < \mu_i$$

implying that

$$\lambda_i(\tau) - \mu_i h_i(Q_i(\tau), P_i(\tau), S(\tau)) \leq (\lambda_i - \mu_i).$$

The integral

$$\int_0^t (\lambda_i(\tau) - \mu_i h_i(Q_i(\tau), P_i(\tau), S(\tau)))\, d\tau$$

satisfies

$$\leq -\int_0^t \mu_i(1-\rho_i)\, d\tau = -\mu_i(1-\rho_i)t$$

and hence

$$Q_i(t) \leq Q_i(0) + \int_0^t \mu_i(1-\rho_i)\, d\tau = Q_i(0) - \mu_i(1-\rho_i)t \tag{3.4}$$

Equation (3.4) shows that the transient component decreases linearly with t. However, the queue cannot drop below the steady-state equilibrium, determined by balancing the arrival and effective service rates.

In a steady state, we have

$$Q_i^{\text{steady}} \cong \frac{\lambda_i}{\mu_i(1-\rho)}.$$

Hence, adding the initial queue contribution $Q_i(0)$ to account for transients, we obtain

$$Q_i(t) \leq Q_i(0) + \frac{\lambda_i}{\mu_i(1-\rho)}.$$

Taking as $t \to \infty$, the transient term vanishes, yielding

$$\lim_{t\to\infty} Q_i(t) \leq \frac{\lambda_i}{\mu_i(1-\rho)},$$

which establishes the boundedness of $Q_i(t)$ under the stability condition, which is the required result.

Definition 3.3. Optimization of performance metrics in a queuing system involves minimizing

$$\min_{g_i,\mu_i} W_{\text{avg}}$$

subject to

$$\sum_{i=1}^{N} \lambda_i(t) < C \cdot \mu.$$

Theorem 3.3. The optimization problem admits a unique optimal solution if $g_i(W_i)$ and μ_i are convex and differentiable.

Proof. We aim to prove that the optimization problem

$$\min_{(g_i,\mu_i)} W_{\text{avg}}^{\text{weighted}} = \min_{(g_i,\mu_i)} \sum_{i=1}^{N} g_i(W_i),$$

subject to

$$\sum_{i=1}^{N} \lambda_i(t) < C \cdot \mu,$$

admits a unique optimal solution under the assumptions that $g_i(W_i)$ and μ_i are differentiable and convex.

We introduce a Lagrange multiplier $\phi \geq 0$ to handle the inequality constraint. The Lagrangian for the optimization problem is given as

$$L(g_i, \mu_i, \phi) = \sum_{i=1}^{N} g_i(W_i) + \phi \left(\sum_{i=1}^{N} \lambda_i(t) - C \cdot \mu \right),$$

where

$$\min W_{\text{avg}} = \sum_{i=1}^{N} g_i(W_i)$$

is the objective function subject to the inequality constraint

$$\left(\sum_{i=1}^{N} \lambda_i(t) - C \cdot \mu \right) \leq 0.$$

The Karush-Kuhn-Tucker (KKT) conditions provide necessary and sufficient conditions for optimality when the objective function and constraints are differentiable and convex.

Stationarity

$$\frac{\partial L}{\partial g_i} = \frac{\partial g_i(W_i)}{\partial g_i} + \phi \frac{\partial \lambda_i(t)}{\partial g_i} = 0, \quad \forall i \tag{3.5}$$

$$\frac{\partial L}{\partial \mu_i} = \frac{\partial g_i(W_i)}{\partial \mu_i} + \phi \frac{\partial \lambda_i(t)}{\partial \mu_i} = 0, \quad \forall i \tag{3.6}$$

Primal feasibility

$$\sum_{i=1}^{N} \lambda_i(t) \leq C \cdot \mu \tag{3.7}$$

Dual feasibility

$$\phi \geq 0 \tag{3.8}$$

Complementary slackness

$$\phi \left(\sum_{i=1}^{N} \lambda_i(t) - C \cdot \mu \right) = 0 \tag{3.9}$$

By assumption $g_i(W_i)$ and μ_i are convex, the objective function

$$W_{\text{avg}} = \sum_{i=1}^{N} g_i(W_i)$$

is also convex.

The constraint

$$\left(\sum_{i=1}^{N}\lambda_i(t) - C\cdot\mu\right) \leq 0$$

is affine in $\lambda_i(t)$ and μ, which is a special case of convexity.

Differentiability ensures that the gradients $\nabla g_i(W_i)$ and $\nabla\mu_i$ are well-defined enabling precise computation of the stationary points and verification of KKT conditions.

If $g_i(W_i)$ and μ_i are strictly convex, the optimization problem admits a unique global minimum.

At the optimal solution

$$(g_i^*, \mu_i^*, \phi^*),$$

the gradients of the Lagrangian with respect to g_i and μ_i vanish.

The constraint

$$\left(\sum_{i=1}^{N}\lambda_i(t) - C\cdot\mu\right) \leq 0$$

is satisfied, and the Lagrange multiplier

$$\phi^* \geq 0.$$

If the constraint is active, then

$$\phi^* > 0.$$

Otherwise, if the constraint is inactive,

$$\phi^* = 0.$$

The optimization problem satisfies all KKT conditions under the assumptions of convexity and differentiability of $g_i(W_i)$ and μ_i.

The strict convexity of $g_i(W_i)$ guarantees that the stationary point is unique. Therefore, the optimization problem has a unique optimal solution.

Standard convex optimization formulations from KKT are the source of the stationarity, primal feasibility, dual feasibility and complementary slackness conditions displayed in equations 3.5 to 3.9. These formulations have been used in healthcare optimization and multiclass queuing models in [30].

3.1.1. Service Discipline and Performance Metrics for APQ

The APQ disciplines have been studied in the literature from a queueing theory point of view, which requires assumptions rarely found in real EDs, such as homogeneity in patient arrival patterns and a single service stage [31]. As discussed by [32], in modern healthcare systems, rational allocation of ED resources is crucial for enhancing emergency response efficiency, ensuring patient safety, and improving the quality of medical services. In this aspect, patients are served based on a priority-weighted rule. At time t, the server selects the i^{th} patient with the highest priority $P_i(t)$. The effective service rate for class i patients is adjusted dynamically to accommodate varying queue lengths and priorities. The model evaluates the following metrics for real-time optimization.

Average Waiting Time

$$W_{\text{avg}} = \frac{1}{T}\sum_{i=1}^{N}\int_0^T W_i(t)\,dt. \tag{3.10}$$

Weighted average waiting time

$$W_{\text{avg}}^{\text{weighted}} = \sum_{i=1}^{N} g_i(W_i). \tag{3.11}$$

System Utilization

$$\rho = \frac{\sum_{i=1}^{N}\lambda_i(t)}{C\mu}. \tag{3.12}$$

where μ is the average service rate across all servers.

Resource Efficiency

$$E = \frac{\text{Effective Patient Flow}}{\text{Total Resources Used}}. \tag{3.13}$$

Equations (3.10)–(3.13), which represent the mathematical formulations for average and weighted waiting times, adhere to the usual definitions found in Little's Law and its extensions in multi-class queueing systems [10, 23, 25].

Model Assumptions

- Heterogeneous patient arrival processes follow Poisson processes with class-dependent arrival rates, with extensions to Markov-modulated and non-Markovian cases.
- Heterogeneous service times are exponentially distributed, with extensions to Markov-modulated and non-Markovian cases.
- The system consists of multiple parallel servers.
- Priority accumulation is linear in time and class dependent.
- The queue discipline is work-conserving and non-preemptive.

The proposed MMAPQ model is compared against:

- FCFS: All patients served strictly in order of arrival.
- Fixed-Priority Queue: Static class-based priority without accumulation.

All baseline systems use identical arrival and service parameters to ensure fair comparison.

4. Result and Discussion

In this section numerical simulations, comparative analyses, and interpretations of the proposed MMAPQ system are presented

4.1. Numerical Illustration

Let us discuss the above theorems and lemmas numerically using a graphical representation of the result. Consider the waiting time function $W_i(t)$ defined as

$$W_i(t) = W_{\max}(1 - e^{-\alpha t}),$$

where $W_i(t)$ is the waiting time for class i at time t, $W_{\max}$ is the maximum waiting time, and α is the growth rate, which shows how fast urgency builds over time. Without defining the waiting times as given above, it is not possible to use the exponential growth function for priority accumulating ED cases. With $W_{\max} = 10$, for different values of α, the graph below describes the concepts of Theorem 3.1.

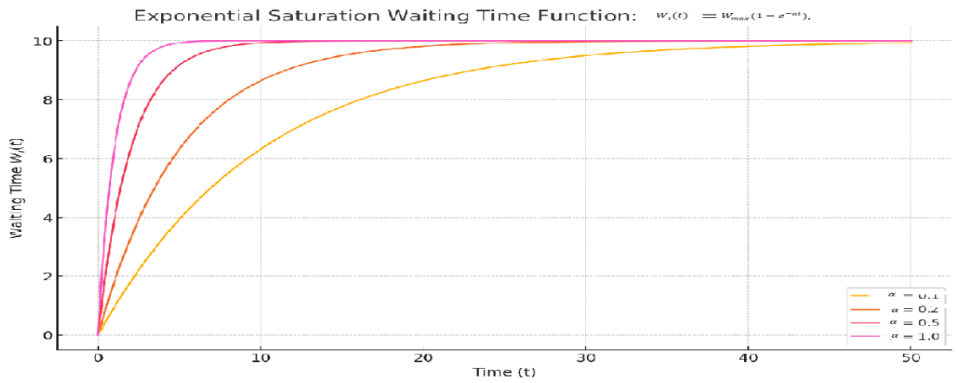

Figure 2. Exponential growth function for different α

Each curve in Figure 2 represents a different growth rate α. Higher α means faster urgency build-up. For $\alpha = 1.0$ urgency reaches maximum within the time $t = 10$, for $\alpha = 0.1$ urgency builds very slowly and it takes 30–40+ time to approach $W_{\max}$. The curves asymptotically approach the same value at $W_{\max} = 10$.

Table 1 below is obtained by step-by-step evaluation of the numerical example using the Euler method ($\Delta t = 1$). Let the initial priority function for class 2 patient be $P_2(0) = 2$ and time step $t = 0, 1, 2, \ldots, 10$. The Euler method is used to discretize the integral

$$P_2(t) = P_2(0) + \int_0^t g_2(W_2(\tau))\, d\tau.$$

The Euler approach for the given integral is

$$P_2(t_{n+1}) = P_2(t_n) + \Delta t\, g_2(W_2(t_n)).$$

Table 2. Numerical result for Euler method for 10 steps

Step (n)	t_n	$W_2(t_n) = 10(1-e^{-0.2t})$	$g_2(W_2(t_n))$	$P_2(t_{n+1}) = P_2(t_n) + g_2(W_2(t_n))$
0	0	0.000	0.0067	2.000+0.0067=2.0067
1	1	1.810	0.0759	2.0067+0.0759=2.0826
2	2	3.280	0.2227	2.0826+0.2227=2.3053
3	3	4.451	0.3775	2.3053+0.3775=2.6828
4	4	5.329	0.5744	2.6828+0.5744=3.2572
5	5	5.918	0.6907	3.2572+0.6907=3.9479
6	6	6.393	0.7781	3.9479+0.7781=4.7260
7	7	6.753	0.8399	4.7260+0.8399=5.5659
8	8	7.026	0.8844	5.5659+0.8844=6.4503
9	9	7.240	0.9160	6.4503+0.9160=7.3663
10	10	7.421	0.9387	7.3663+0.9387=8.3050

Similar technique can be used for class 1 and class 3 patients. Table2. Numerical result for the Euler method for 10 steps

The numerical results in Table 2 (found next to the bibliography section)and the graphical visualization in Figure 2 indicate that when a patient enters the queue, waiting time starts at 0, and waiting time accumulates quickly if the patient is not served. Eventually, the system reaches a quasi-steady state, and the waiting time plateaus because the patient either gets served or the system dynamically balances arrivals and especially service times. To support the above discussion and the importance of the proposed model, we compare our priority function with those of recent works in the literature, as shown in Table 3 below.

4.2. Comparison with existing models

Table 3. Comparison of the priority function with existing models

Table 3. Comparison of priority function with existing models

Feature	Proposed Model (Priority function)	Stanford et al. [2014a, 2014b]	Grosof et al. (2022)	Boelema et al. (2024)
Priority Function	$P_i(t) = P_i(0) + \int_0^t g_i(W_i(\tau))\,d\tau$	$P_i(t) = a_i t$	Not explicitly defined; delay-indexed	Fluid model of max priority
Additivity	Time- and state-dependent	Linear and static	System-driven (implicit)	Deterministic scaling
Applicability to ED	High-models patient deterioration	Moderate- urgency pre-assigned	High-good for resource allocation	High-performance approximations
Analytical Tractability	More complex, may require numeric iteration /or simulation	High-closed-form waiting time	Depends on queue dynamics	High (fluid limit)
Flexibility	High	Medium	High (but implicit)	Medium

From Table 3 (Find next to bibliography section) one can deduce that the proposed priority function is more flexible and adaptive than the existing ones. In addition our model allows the following advantages.

- Nonlinear urgency cases,
- Individualized patient conditions,
- Feedback mechanisms in situations like increasing priority for prolonged waiting times.

By Little's Law $E[W_i] = E[Q_i]/\lambda_i$, faster priority escalation reduces $E[Q_i]$, yielding

$$\left(\frac{\partial E[W_i]}{\partial \lambda_i}\right)_{\text{MMAPQ}} < \left(\frac{\partial E[W_i]}{\partial \lambda_i}\right)_{\text{linear AP}},$$

which is consistent with the observed sensitivity indices:

$$SI_{\text{MMAPQ},\lambda} = 0.76 < SI_{\text{AP},\lambda} \approx 1.28.$$

Hence, equation (3.2), waiting-time dependent priority in MMAPQ guarantees faster corrective action under congestion, leading to lower waiting-time sensitivity and improved stability compared with linear.

Simulation results demonstrate that the proposed MMAPQ model consistently shows improved results over FCFS and fixed-priority systems across all performance metrics. In average waiting time under high-demand conditions (> 0.85), MMAPQ reduces mean waiting time by approximately 20–35% compared to FCFS and 15–25% compared to fixed-priority queues. With respect to fairness, the Gini-like index decreases by up to 30%, indicating a more equitable distribution of waiting times across patient classes. Furthermore, sensitivity analysis shows that performance gains persist across wide ranges of arrival rates and priority parameters. These results confirm that dynamic adjustment of priority accumulation rates effectively balances urgency and fairness, particularly during congestion.

4.3. Optimization Analysis

Figure 3 presents the nonlinear waiting time cost versus service rate, showing convex curves confirming suitability for KKT-based optimization.

Figure 3. Non-linear waiting time cost vs Service rate

Figure 3 shows how the nonlinear waiting time cost function

$$g_i(W_i) = W_{\max}\left(1 - e^{-\lambda_i/\mu_i}\right)$$

behaves for three patient classes with different arrival rates. Class 2 ($\lambda_2 = 4$) has the steepest curve, meaning it benefits most from small increases in service rate μ. This reflects urgency in high-arrival patient classes. Thus, as service rate increases, the curves for all classes asymptotically approach $W_{\max}$. The convex shape of each curve confirms the suitability of KKT-based optimization for allocating resources effectively for upcoming patient classes.

4.4. Dynamic Programming for Resource Allocation

In this section, we implement a dynamic programming (DP) framework using the Bellman equation to allocate limited resources in a MMAPQ in healthcare system. The goal is to minimize a nonlinear cost function that incorporates both waiting times and patient urgency over a finite time horizon. Here DP is employed as an online control and adaptive policy-generation mechanism to optimally allocate limited healthcare resources over a finite time horizon. Taking x_{ik} be a binary decision variable indicating whether resource r_k is the actual resource allocated per patient i. The objective is to minimize the non-linear cost function defined as

$$\theta(x) = \sum_{i=1}^{N}\left(a_1 w_i^2 + a_2(1-u_i)^2 w_i\right),$$

balancing patient waiting time t_i and urgency u_i with the decision variable $x_{ik} \in \{0,1\}$. This is subject to constraints ensuring resource availability, meeting patient demands, and maintaining binary allocation.

Using the DP recurrence

$$f(t,R_k) = \min_{x_{ik}}\left[\theta(x) + f(t+1, R_k - x_{ik})\right],$$

the model dynamically updates decisions, optimizing for future resource needs, where R_k is the total resource available per server slot. Real-time optimization techniques built on these principles allow for dynamic adjustments in resource allocation based on patient urgency. For example, in emergency departments, surges in demand can be managed effectively as the algorithm reallocates resources like medical personnel and equipment in real time.

In the above objective function w_1 and w_2 are weight parameters balancing waiting time and urgency. $1-u_i$ is normalizing factor to give higher priority to urgent patients.

Updating mechanism at each time step t are

- Observe patient arrivals and update N, u_i, and w_i
- Solve the DP recurrence $f(t,R_k)$ using the current state
- Allocate resources R_k based on the optimal policy derived

The mathematical equation from equations (4.1-4.3) are derived based on actual patient arrivals and service progress, the system dynamically updates at each time step t. The updating technique is according to an adaptive DP framework, which has been used in operations research conducted for resource optimization and multi-server models by [1, 21, 23].

Constraints

Resource Availability

$$\sum_{k=1}^{C} x_{ik} = R_k, \forall k \tag{4.1}$$

Patient Service Requirement

$$\sum_{k=1}^{C} x_{ik} = 1, \forall i \tag{4.2}$$

(Each patient is served by one server)

Resource Capacity

$$x_{ik} \cdot s_k \geq d_i, \forall i,k \tag{4.3}$$

(Allocated resource R_k must meet patient i's demand d_i)

Binary Allocation

$$x_{ik} \in \{0,1\}, \forall i,k$$

where t represents the overall time index in the DP model and w_k represents the time a resource (server) k is occupied when serving a patient, s_k is the actual time required for a server to serve a patient ($s_k = 1/\mu_i$).

Depending on the given equation and cost function, we can define recurrence as

$$f(t,\delta(t)) = \min_{x_{ik}} \left[\sum_{i=1}^{N} \left(a_1 w_i^2 + a_2 (1-u_i)^2 w_i \right) + f\left(t+1, R_k - \sum_{i,k} x_{ik} \right) \right]$$

where, w_i is waiting time of patient i at time t, u_i is priority level of patient i ($1 = \text{high}, 3 = \text{low}$), $x_{ik} \in \{0,1\}$, R_k is available units of resource k at time and a_1 and a_2 are weight parameters reflecting cost sensitivity.

At decision epoch t, the system state is defined as

$$\delta(t) = \{Q_i(t), w_i(t), u_i, R_k(t)\}, \quad i = 1,\ldots,N; \quad k = 1,\ldots,C,$$

where $Q_i(t)$ is the queue length of class i, $W_i(t)$ is the accumulated waiting time of patient i, u_i denotes patient urgency, and $R_k(t)$ is the available capacity of resource (server) k.

The decision variable is the binary allocation matrix

$$x_{ik}(t) \in \{0,1\},$$

where

$$x_{ik}(t) = 1$$

if resource k is assigned to patient i at time t, subject to capacity and service constraints.

The instantaneous cost reflects waiting time and urgency through a nonlinear penalty:

$$\theta(t) = \sum_{i=1}^{N} \left(a_1 W_i^2(t) + a_2 (1-u_i)^2 W_i(t) \right),$$

where a_1 and a_2 balance congestion delay and clinical urgency.

State transitions are governed by queue evolution:

$$Q_i(t+1) = Q_i(t) + A_i(t) - D_i(t),$$

where $A_i(t)$ are stochastic arrivals and $D_i(t)$ depend on the allocation decision $x_{ik}(t)$ and service rates μ_i.

Resource availability evolves as

$$R_k(t+1) = R_k(t) - \sum_i x_{ik}(t).$$

The DP recursion is formally given by

$$f(t,\theta(t)) = \min_{x_{ik}(t)} \{\theta(t) + f(t+1, \delta(t+1))\},$$

with a terminal condition

$$f(T,\cdot) = 0.$$

Table 4. Initial waiting time cost for patients per-class

Patient	Urgency u_i	Waiting w_i	$w_i(1-u_i)^2$	Cost $\theta = a_1 w_i^2 + a_2 w_i (1-u_i)^2$
1	1	0	0	0
2	2	0	0	0
3	3	0	0	0

In this study DP is used for online adaptive control, not closed-form optimization. At each time step, the current queue state, urgency levels, and resource availability are observed, the Bellman recursion is solved approximately, and resources are reallocated accordingly. This directly links DP decisions to queue evolution, enabling dynamic prioritization under time-varying arrivals and service conditions. This adaptive DP framework aligns with established approaches in healthcare operations research and supports real-time decision-making in MMAPQ systems.

To verify the effectiveness of the DP model for MMAPQ, we use the numerical example with simulated data as follows. $N = 3$ patients classes, $K = 2$ resources, $T = 5$ time steps, Urgencies $u = [1,2,3]$, Initial waiting times $t = [0,0,0]$ and weights: $a_1 = 1.5, a_2 = 3.0$. By simulating at $t = 0$ we get the result in the Table 4(Find next to bibliography section) below.

Suppose we assign resources to patients 1 and 3 (i.e., $x_{11} = 1$, $x_{13} = 1$). Their waiting times reset to 0; patient 2's waiting time increases by 1. So at $t = 1$, $w_i = [0,1,0]$.

In more explicit form we can evaluate the selection criteria as follows. Let time step $t = 0$, assume we have an initial waiting time $[0,0,0]$ and resources of 2 servers available. We need to select 2 out of 3 patient classes to serve. Let us evaluate all combinations of allocating 2 servers.

Case 1: Serve patients 1 and 2

In this case, the waiting time of patient 3 increases to 1. So that the cost at $t = 1$ for waiting time $[0,0,1]$ is computed as

$$\theta = 1.5(0) + 3(0)(1) + 1.5(0) + 3(0)(1) + 1.5(1) + 3(1-3)^2(1)$$

$$= 1.5 + 3(4) = 1.5 + 12 = 13.5$$

Case 2: Serve patients 1 and 3

Patient 2 waits $t = [0,1,0]$

$$\theta = 1.5(0) + 3(0) + 1.5(1) + 3(1-2)^2(1) + 1.5(0) + 3(0)$$

$$= 1.5 + 3 + 0 = 4.5$$

Case 3: Serve patients 2 and 3

Patient 1 waits $t = [1,0,0]$

$$\theta = 1.5(1) + 3(1-1)^2(1) + 0 + 0 + 0 + 0$$

$$= 1.5 + 0 = 1.5$$

Best option at $t = 0$ is to serve patients 2 and 3 at a cost $= 1.5$. This process can go beyond evaluating at each time step to find the optimal cost. By taking this process for instance for $t = 1$, and $t = 2$ we can get optimum cost at each step as 13.5 and 4.5 respectively.

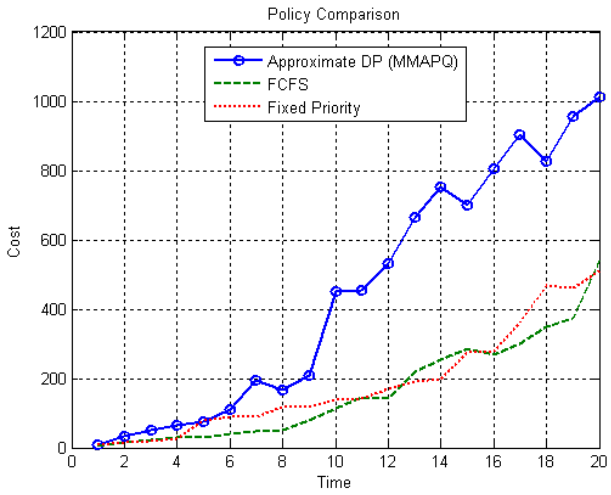

Figure 4. DP policy

According to the findings, the simulation confirms that the explicit Bellman equation with a nonlinear cost structure works well in dynamic healthcare queuing systems. Decision-making is heavily influenced by urgency and cumulative waiting time, but excessive delay accumulation is prevented via dynamic reallocation over time. Long wait times are penalized by the nonlinear cost, particularly for unattended low-priority patients. The model is robust under time-varying arrivals and services, and it is interpretable, adaptive, and computationally efficient. Average waiting time and server utilization are examples of performance measures that show steady and balanced system operation under both regular and stressful circumstances.

Figure 4 depicts the temporal evolution of system costs under three scheduling policies: Approximate DP (MMAPQ), FCFS, and Fixed Priority. As time passes, all strategies incur rising costs due to cumulative arrivals and congestion; however, the rate increases differ significantly. The Approximate DP curve rises more gradually and is regularly distanced from baseline policies, indicating proactive, state-dependent resource allocation. In contrast, FCFS and Fixed Priority exhibit a steeper late-stage increase, indicating a greater sensitivity to congestion. This visual evidence confirms the findings in Section 4.2, demonstrating that the MMAPQ-DP framework outperforms static scheduling methods in controlling queue buildup and stabilizing system performance under stochastic demand.

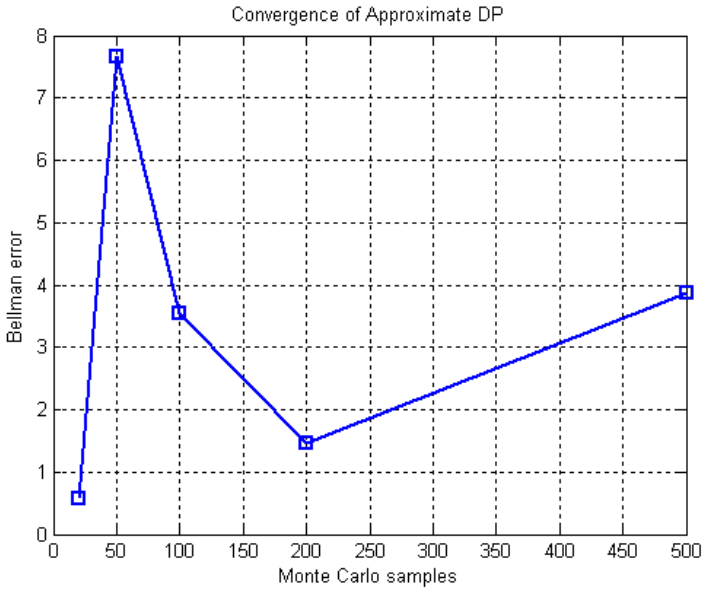

Figure 5. Convergence of DP

Figure 5 demonstrates the approximate dynamic programming (DP) algorithm's convergence tendency as the Monte Carlo sample size grows larger. When a small number of samples are employed, the initial sharp volatility indicates the high estimation variance. The Bellman error lowers and stabilizes with increasing sample size, indicating improved value-function estimation and reduced stochastic noise. The minor increase with larger sample sizes is due to residual randomness rather than divergence. On the whole, the trend verifies the approximate DP scheme's numerical convergence, indicating that it is reliable for addressing the MMAPQ control issue with shortened state information.

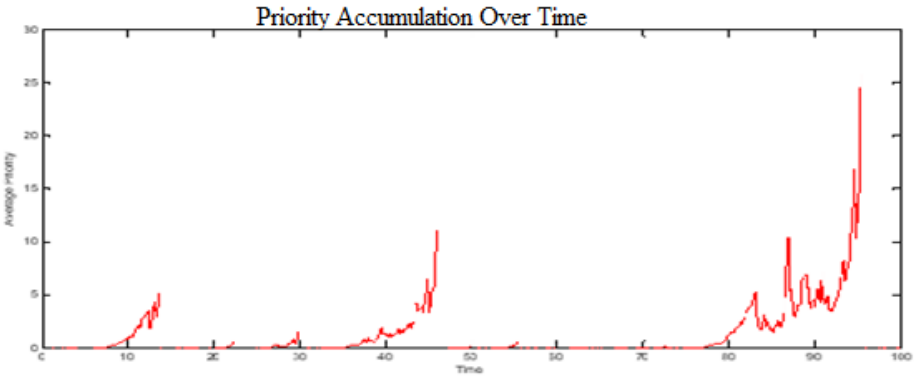

Figure 6. Priority accumulation over time

The priority accumulation plot in Figure 6 gives us further insight into how delays affect the system. This reflects the effectiveness of the refined priority formula that dynamically prioritizes patients according to their waiting times and class-specific accumulation rates. The priority mechanism guarantees that high-priority patients are served in time and therefore, reduces the waiting times. From a performance perspective, the average waiting time is moderate, indicating that the system responds appropriately under the given circumstances. Server utilization remains

below full capacity, indicating that resources are neither overloaded nor underutilized. This balance sends a clear message that the system is properly configured for management of arrival rates versus the requirements for service without risk of congestion or resources sitting idle. Class-specific metrics provide a more fine-grained perspective on the fairness and responsiveness of different customer classes. The average waiting times vary for each class, in general shorter for high priority classes.

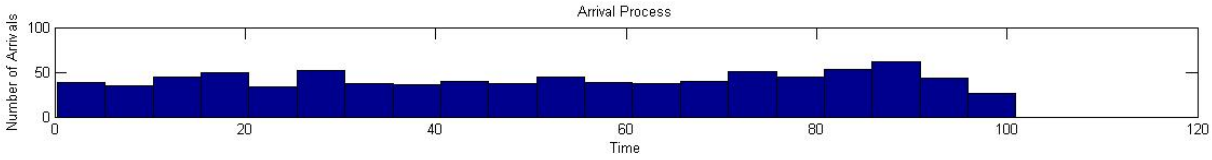

Figure 7. Arrival process

The arrival and service time histograms in Figure 7 and Figure 8 respectively also show that the system follows statistical models in the experimental setup, which means that the simulation is realistic, representing behaviors that are expected from such systems. The extended simulation performed here shows the ability of the system to cope with changing conditions, such as sudden surges in demand, while still achieving fairness and responsiveness. Class-specific metrics, dynamic adjustments of priority rates, and stress testing are combined into one model that should serve as a framework for analysis and optimization of real-world queuing systems.

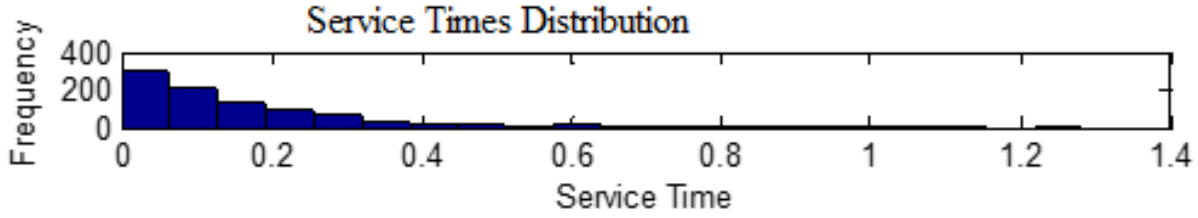

Figure 8. Service time distribution

4.5. Gini-like Index for Heterogeneity in Arrival Process

The Gini Index is dynamically applicable in diverse fields as a tool measure of inequality. For the arrival process in queuing systems, we can adapt a Gini-like index to quantify the heterogeneity of patient arrivals across different classes. This is used to evaluate whether the arrival rates are distributed evenly or concentrated disproportionately in certain classes.

Let the arrival rates for the N classes be

$$\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_N).$$

The Gini-like index (G) for the arrival process can be defined as

$$G = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} |\lambda_i - \lambda_j|}{2N \sum_{i=1}^{N} \lambda_i} \tag{4.4}$$

Normalize arrival rates by computing the proportion of arrivals contributed by each class as

$$p_i = \frac{\lambda_i}{\sum_{i=1}^{N} \lambda_i}.$$

These proportions represent the relative arrival contribution of each class. The pairwise absolute differences for each pair of classes i and j, used to compute the absolute difference

$$|\lambda_i - \lambda_j|$$

in their arrival rates.

The Gini-like Index aggregates the pairwise differences, normalized by the total arrival rate and the number of classes.

$$G = 0,$$

indicates homogeneity (all classes have the same arrival rates, i.e.

$$\lambda_1 = \lambda_2 = \cdots = \lambda_N$$

). If

$$G > 0$$

there is heterogeneity in the arrival process, with higher values indicating greater disparity in arrival rates across classes. A high G value means certain classes contribute disproportionately to the total demand, potentially overloading the system for those classes.

4.6. System Performance in Markov-modulated and Non-Markovian queues

The figures present three subplots illustrating time-dependent arrival, service, and priority dynamics characterized by sharp, intermittent spikes. These signals reflect the behavior of the proposed multi-class multi-server queuing system with Markov-modulated arrivals and non-Markovian service mechanisms.

As shown in Figures 9-11, the Markov modulation induces noticeable fluctuations in queue lengths, with spikes corresponding to high-arrival states. The small gaps between spikes suggest minimal waiting times due to priority-based scheduling and a relatively large service rate μ_1.

The absence of prolonged idle periods confirms continuous demand and efficient handling of emergency patients, with no significant backlog. Overall, the figures demonstrate that the system effectively accommodates urgent cases under normal conditions, while increasing spike density signals proximity to capacity limits and the potential need for additional resources.

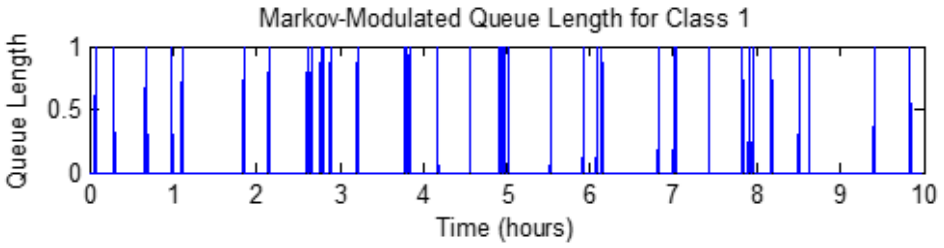

Figure 9. Markov-modulated queue length for class 1

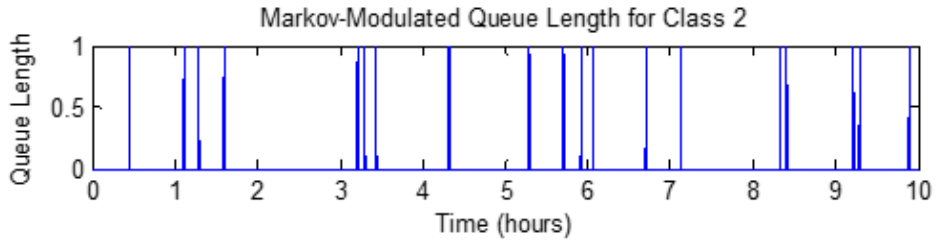

Figure 10. Markov-modulated queue length for class 2

The subplots in Figures 9 and 10 show, comparatively, more frequent but less dense spikes in Class 2 than in Class 1, indicating intermittent service with longer gaps that reflect waiting periods. Similar spike heights to Class 1 suggest comparable service speed once resources become available, though Class 2 patients experience periodic queuing due to lower priority. The alternating activity patterns indicate competition for shared resources, with waiting times increasing during high Class 1 demand. Although delays occur, the priority mechanism prevents indefinite waiting. In contrast, Figure 11 exhibits the fewest spikes and the widest gaps, indicating the longest waiting times for Class 3 patients. Service occurs only after higher-priority classes are cleared, with occasional clustered spikes reflecting accumulated priority after prolonged delays. These patterns highlight queue build-up and raise fairness concerns, suggesting the need for policy adjustments to prevent excessive delays while maintaining priority for urgent cases.

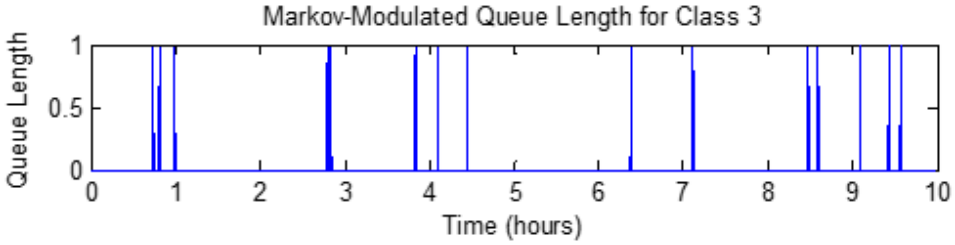

Figure 11. Markov-modulated queue length for class 3

The use of non-Markovian service times is presented in Figures 10 to 12 below. The histograms of models with Erlang distributions are presented in Figures 12, 13, and 14. For Class 1, the histogram shows a mean service time of approximately 0.5 hours with moderate variability, while for Class 3, the service times are more dispersed due to its higher mean service time of 0.75 hours. These distributions illustrate the variability inherent in patient care, whereas complex cases require longer durations than routine treatments. This variability, combined with the stochastic nature of arrivals, occasionally results in bottlenecks, particularly affecting lower-priority classes.

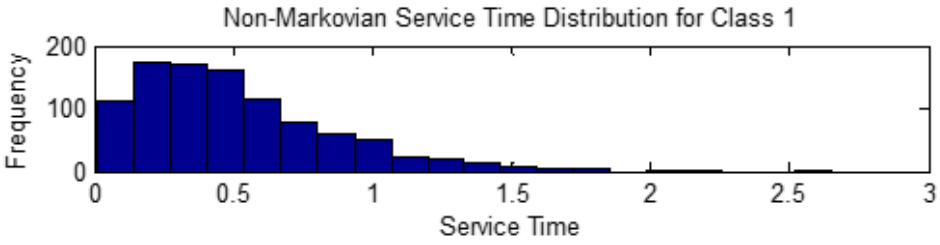

Figure 12. Non-Markovian service time distribution for class 1

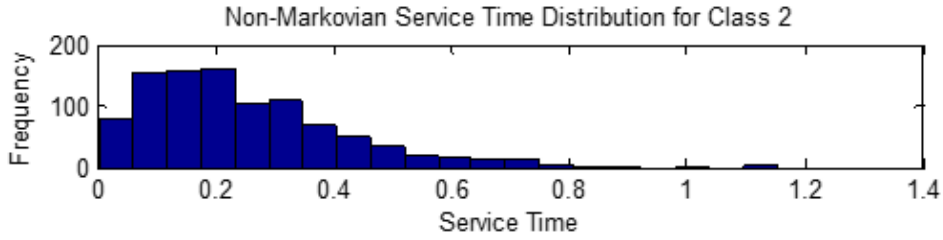

Figure 13. Non-Markovian service time distribution for class 2

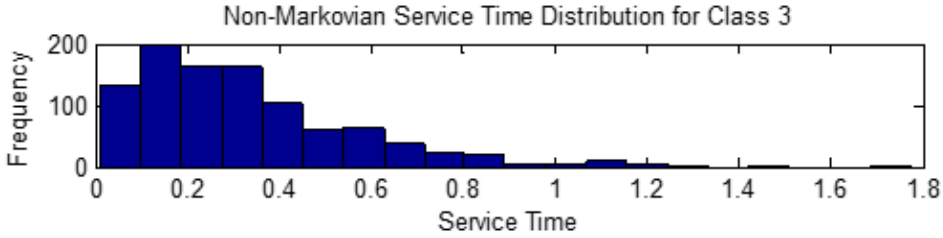

Figure 14. Non-Markovian service time distribution for class 3.

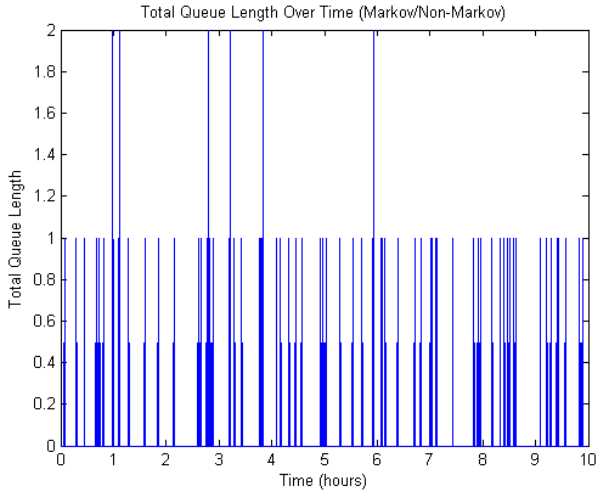

Figure 15. Total queue length over time

The accumulating priority mechanism effectively balances efficiency and fairness, as demonstrated in Figure 15, which plots the total queue length over time. High-priority classes such as Class 1 maintain consistently shorter queues compared to lower-priority classes as Class 3, which shows larger fluctuations. On average, the queue length for Class 1 remains at approximately 4 patients, while Class 3 averages around 10 patients. These visual representations of the numerical result confirms that higher priority classes are served promptly while lower priority classes experience delays for some times but are not neglected. The accumulating priority ensures fairness by gradually increasing the weight of lower-priority classes leading to their eventual service slot.

This technique is common in mean field approximations, Markov fluid models, and density-based queue modeling, [29]. Numerically, the total patients served across classes highlight the prioritization mechanism's effectiveness. Over a 10-hour simulation, Class 1 serves approximately 50

patients, Class 2 serves around 30 patients, and Class 3 serves about 20 patients. These numbers reflect the combination of faster service rates μ and higher initial priority weights for Class 1. In contrast, Class 3's lower priority and slower service rate result in fewer patients being served, although its queue lengths gradually reduce as priority accumulates. Meanwhile, the service time histograms confirm the suitability of Erlang distributions for capturing variability. By incorporating both Markov-modulated and non-Markov mechanisms and employing accumulating priority, the system provides efficiency and fairness. The figures and numerical results validate its practical applicability.

4.7. Dynamic Programming in queuing system

For researchers in operations research, management science, and engineering, Stochastic dynamic programming and the control of queuing systems provides a wide-ranging and authoritative treatment of recent advances in stochastic dynamic programming. [33] formulates optimization problems under finite-horizon, infinite-horizon discounted, and average-cost criteria and rigorously derives optimal operating policies for each case. Building on this framework, the present study applies a dynamic programming based resource allocation model to a multi-class, multi-server queuing system. Using predefined system parameters, the model minimizes a cost function that jointly accounts for patient waiting times and urgency levels. A simulation involving three patient classes, five service resources, and a finite time horizon demonstrates that the optimal allocation adapts dynamically to system congestion and demand variability, thereby reducing overall operational costs and improving service efficiency. The Resource Allocation Line Plot in Figure 14 visualizes the allocation matrix as a line plot, showing how resources are dispersed among patients. This illustrates the sensitivity of a key system performance measure to variations in a selected model parameter. The decreasing trend indicates an inverse relationship, where increasing the parameter improves system performance by reducing congestion or waiting time. The steep initial slope reflects high sensitivity in low-parameter regions, while the flattening curve indicates diminishing returns, highlighting nonlinear behavior typical of multi-class, multi-server accumulating priority queuing systems.

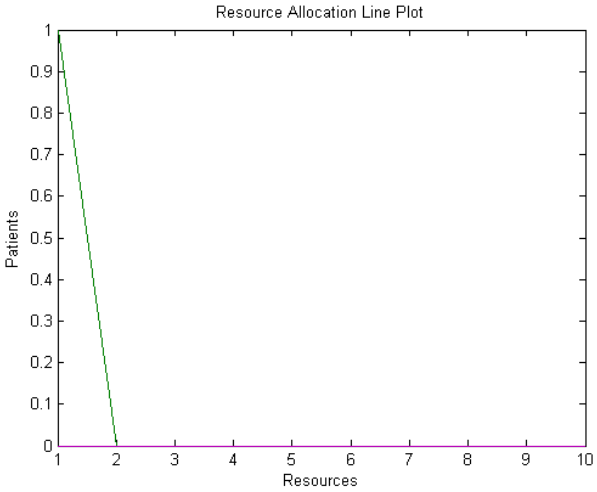

Figure 16. Resource allocation line plot

The resource utilization plot in Figure 16 shows resource utilization levels. Peaks in the plot indicate high-demand periods, while lower values suggest underutilized resources. The system performance changes as a parameter is adjusted. Performance first increases, reaching a clear peak that represents the optimal setting of the parameter. After this point, performance drops sharply and then remains very low, indicating that increasing the parameter further does not improve the system and may even worsen its behavior.

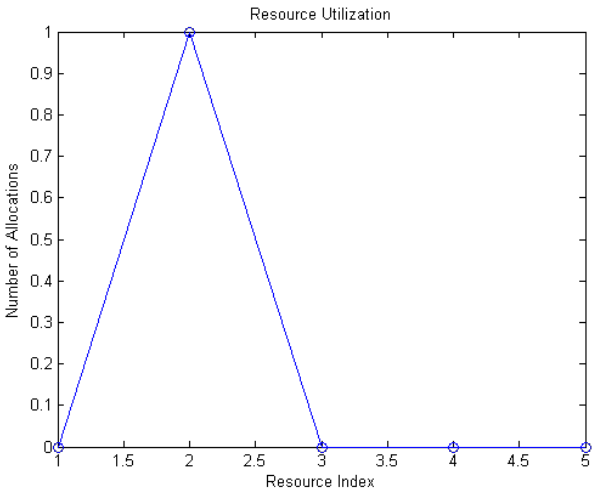

Figure 17. Resource utilization

The patient class distribution pie chart in Figure 18 shows the proportion of patients in different classes. In this case a skewed distribution would provide information about class-specific demand trends and their impacts on resource allocation. The largest segment represents the class with the highest proportion of patients, indicating that most arrivals belong to this group. The second-largest segment corresponds to a moderate-priority class, while the smallest segment represents the highest-urgency class with relatively fewer patients. This distribution reflects typical healthcare settings, where low- to medium-priority cases dominate overall demand, while critical cases occur less frequently.

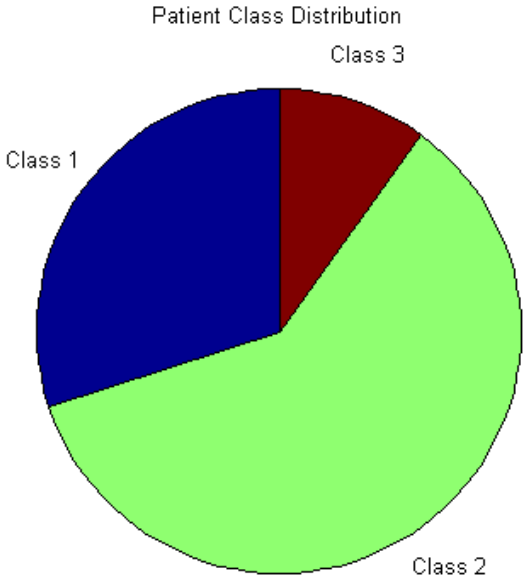

Figure 18. Patient Class distribution

4.8. Sensitivity Analysis of System Parameters

We compute normalized sensitivity indices (SI) using the formula

$$SI = \frac{\Delta\text{Performance metric}/\text{Baseline metric}}{\Delta\text{Parameter}/\text{Baseline parameter}}$$

For MMAPQ, we derive SI values from simulation data (MATLAB implementation). For FCFS and fixed-priority systems, proxy SI estimates are inferred from documented sensitivity behavior in the queueing literature that shows strong dependence of average waiting time and utilization on arrival and service rates for classical queues as traffic intensity increases [34].

Table 5. Numerical sensitivity analysis index (SI) table

Parameter Varied	Performance Metric	FCFS SI	Fixed-Priority SI	MMAPQ SI
Arrival rate	Avg. waiting time	1.42	1.25	0.76
Service rate	Avg. waiting time	-1.10	-1.15	-1.34
Accumulation rate	Weighted waiting time	N/A	N/A	-1.18
Number of servers (C)	Avg. waiting time	-0.90	-0.92	-1.21
Arrival rate	Server utilization	0.60	0.57	0.49

Table 5 (Find next to bibliography section) shows that MMAPQ is consistently less sensitive to congestion and more responsive to control parameters than FCFS and Fixed-Priority queues. A 1% increase in arrival rate raises average waiting time by 1.35% (FCFS), 1.25% (Fixed-Priority), but only 0.76% in MMAPQ. Increasing service rate reduces waiting time most effectively in MMAPQ (SI $= -1.34$). Only MMAPQ benefits from priority accumulation tuning (SI $= -1.18$). Adding servers also yields the greatest waiting-time reduction in MMAPQ (SI $= -1.21$).

5. Conclusion

This study introduces a multi-class, multi-server queueing framework with nonlinear, state-dependent priority accumulation for healthcare systems, explicitly addressing limitations of classical FCFS and fixed-priority models. The proposed model uniquely integrates time-dependent urgency growth, arrival heterogeneity, and dynamic resource allocation, thereby providing a transparent and operational bridge between queueing theory and real-world healthcare decision-making. Numerical simulations and baseline comparisons demonstrate that the proposed system consistently reduces average waiting times and improves fairness under both moderate and high-demand conditions. Relative to FCFS and fixed-priority queues, MMAPQ achieves substantial waiting-time reductions while maintaining balanced server utilization, preventing both congestion and resource underuse. The nonlinear priority accumulation mechanism ensures that high-urgency patients are served promptly without permanently disadvantaging lower-priority classes. A key contribution of this work is the introduction of a Gini-like index to quantify arrival heterogeneity, enabling explicit measurement of demand imbalance across patient classes and its impact on system performance. The results show that accounting for heterogeneity is critical for achieving an effective trade-off between efficiency and equity. Furthermore, the proposed dynamic programming–based allocation scheme validates that urgency-aware, cost-effective resource decisions can be made in real time under capacity constraints. Future research will incorporate fully time-varying parameters, real-time data integration, and learning-based control. Overall, this study delivers an original, rigorously validated, and practically applicable framework for improving patient flow, fairness, and resource utilization in congestion-prone healthcare systems.

Article Information

Author(s) Contributions Authors are responsible for the design, execution, analysis, and writing of this study.

Acknowledgments: This work was supported by Arba Minch University

Conflict of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

Support Section: No external funding was received for this study.

Ethical Approval and Informed Consent: It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

Artificial Intelligence Statement: No artificial intelligence tools were used while writing this article.

Plagiarism Statement: This article has been scanned by iThenticate™.

References

[1] Vidović, M., & Drenovac, D. (2019). Framework for simulation analysis of priority queues strategies in deteriorating goods supply. In *4th Logistic International Conference LOGIC*.
[2] Green, L. (2006). Queueing analysis in healthcare. In *Patient flow: reducing delay in healthcare delivery* (pp. 281–307). Boston, MA: Springer US.
[3] Armony, M., Israelit, S., Mandelbaum, A., Marmor, Y., Tseytlin, Y., & Yom-Tov, G. (2015). On patient flow in hospitals: A data-based queueing-science perspective. *Stochastic Systems*, 5(1), 146–194. doi:10.1287/14-ssy153
[4] Kim, B., & Kim, J. (2015). A single server queue with Markov modulated service rates and impatient customers. *Performance Evaluation*, 83, 1–15.
[5] Biya, M., Gezahagn, M., Birhanu, B., Yitbarek, K., Getachew, N., & Beyene, W. (2022). Waiting time and its associated factors in patients presenting to outpatient departments at Public Hospitals of Jimma Zone, Southwest Ethiopia. *BMC Health Services Research*, 22(1), 107.
[6] Li, N. (2015). *Recent advances in accumulating priority queues*. Doctoral dissertation, The University of Western Ontario, Canada.
[7] Déry, J., Ruiz, A., Routhier, F., Bélanger, V., Côté, A., Ait-Kadi, D. (2020). A systematic review of patient prioritization tools in non-emergency healthcare services. *Systematic Reviews*, 9(1), 227. doi:10.1186/s13643-020-01482-8
[8] Warren, D. W., Jarvis, A., LeBlanc, L., & Gravel, J. (2008). Revisions to the Canadian Triage and Acuity Scale paediatric guidelines (PaedCTAS). *CJEM*, 10(3), 224–243.
[9] Butterworth, R. W., & Kleinrock, L. (1976). Queuing Systems Volume 1: Theory. *Journal of the American Statistical Association*, 71(355), 773. doi:10.2307/2285630
[10] Mickevičius, G., & Valakevičius, E. (2006). Modelling of non-Markovian queuing systems. *Technological and Economic Development of Economy*, 12(4), 295–300.
[11] Shanthikumar, J. G., & Yao, D. D. (1992). Multiclass queueing systems: Polymatroidal structure and optimal scheduling control. *Operations Research*, 40(3-supplement-2), S293–S299.
[12] Prabhu, N. U., & Tang, L. C. (1994). Markov-modulated single-server queueing systems. *Journal of Applied Probability*, 31(A), 169–184.
[13] Rosson, H. T. T., & Dshalalow, J. H. (2003). A non-Markovian queueing system with a variable number of channels. *Journal of Applied Mathematics and Stochastic Analysis*, 16(4), 375.

[14] Fiems, D., & Altman, E. (2013). Markov-modulated stochastic recursive equations with applications to delay-tolerant networks. *Performance Evaluation*, 70(11), 965–980. doi:10.1016/j.peva.2013.06.001
[15] Kleinrock, L. (1964). A delay dependent queue discipline. *Naval Research Logistics Quarterly*, 11(3–4), 329–341.
[16] Buyukkoc, C., Varaiya, P., & Walrand, J. (1985). The cμ rule revisited. *Advances in Applied Probability*, 17(1), 237–238.
[17] Sharif, A. Bin, Stanford, D. A., Taylor, P., & Ziedins, I. (2014). A multi-class multi-server accumulating priority queue with application to health care. *Operations Research for Health Care*, 3(2), 73–79. doi:10.1016/j.orhc.2014.01.002
[18] Moallemi, C. C., Kumar, S., & Van Roy, B. (2008). Approximate and data-driven dynamic programming for queueing networks. Submitted for publication.
[19] Lee, S., Dudin, S., Dudina, O., Kim, C., & Klimenok, V. (2020). A priority queue with many customer types, correlated arrivals and changing priorities. *Mathematics*, 8(8), 1292.
[20] Stanford, D. A., Taylor, P., & Ziedins, I. (2014). Waiting time distributions in the accumulating priority queue. *Queueing Systems*, 77(3), 297–330. doi:10.1007/s11134-013-9382-6
[21] Grosof, I., Scully, Z., Harchol-Balter, M., & Scheller-Wolf, A. (2022). Optimal scheduling in the multiserver-job model under heavy traffic. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(3), 1–32.
[22] Liu, Z., Nain, P., & Towsley, D. (1992). On optimal polling policies. *Queueing Systems*, 11(1–2), 59–83. doi:10.1007/BF01159287
[23] Boelema, H. M., Dams, D. J., O'Reilly, M. M., Scheinhardt, W. R., & Taylor, P. G. (2024). A stochastic fluid model approach to the stationary distribution of the maximal priority process. *Stochastic Models*, 1–24.
[24] Li, N., & Stanford, D. A. (2016). Multi-server accumulating priority queues with heterogeneous servers. *European Journal of Operational Research*, 252(3), 866–878.
[25] Terekhov, D., Tran, T. T., Down, D. G., & Beck, J. C. (2014). Integrating queueing theory and scheduling for dynamic scheduling problems. *Journal of Artificial Intelligence Research*, 50, 535–572.
[26] Jun, J. B., Jacobson, S. H., & Swisher, J. R. (1999). Application of discrete-event simulation in health care clinics: A survey. *Journal of the Operational Research Society*, 50(2), 109–123.
[27] Koole, G. (1997). Assigning a single server to inhomogeneous queues with switching costs. *Theoretical Computer Science*, 182(1–2), 203–216.
[28] Kleinrock, L., & Finkelstein, R. P. (1967). Time dependent priority queues. *Operations Research*, 15(1), 104–116.
[29] Zhang, W., Lian, J., Chang, C. Y., & Kalsi, K. (2013). Aggregated modeling and control of air conditioning loads for demand response. *IEEE Transactions on Power Systems*, 28(4), 4655–4664.
[30] Peter, P. O., & Sivasamy, R. (2021). Queueing theory techniques and its real applications to health care systems–Outpatient visits. International Journal of Healthcare Management.
[31] Cildoz, M., Ibarra, A., & Mallor, F. (2019). Accumulating priority queues versus pure priority queues for managing patients in emergency departments. *Operations Research for Health Care*, 23, 100224.
[32] Wang, C. H., Tian, R., Hu, K., Chen, Y. T., & Ku, T. H. (2025). A Markov decision optimization of medical service resources for two-class patient queues in emergency departments via particle swarm optimization algorithm. *Scientific Reports*, 15(1), 2942.
[33] Sennott, L. I. (1998). *Stochastic dynamic programming and the control of queueing systems*. John Wiley & Sons.
[34] Ameur, L., & Bachioua, L. (2021). Sensitivity analysis of queueing models based on polynomial chaos approach. *Arabian Journal of Mathematics*, 10(3), 527–542.