International Journal of Basic and Applied Sciences, 14 (SI-2) (2025) 199-206

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/y0fwh837
Research paper

Dynamic Queuing Algorithms for Optimized Healthcare Appointment and Patient Flow Management in OPD Systems

M. Kaif Qureshi [1] *, Aniket Pradhan [1], Parth Wande [1], Sarvesh Dongare [1], Dr. Nupur Giri [2], Dr. Gresha Bhatia [3]

[1] Department of Computer Engineering, Vivekanand Education Society's Institute of Technology (Affiliated to the University of Mumbai) Mumbai, India
[2] Head of Department, Department of Computer Engineering, Vivekanand Education Society's Institute of Technology (Affiliated to the University of Mumbai), Mumbai, India
[3] Deputy Head of Department, Department of Computer Engineering, Vivekanand Education Society's Institute of Technology (Affiliated to the University of Mumbai), Mumbai, India
**Corresponding author E-mail:d2021.mkaif.qureshi@ves.ac.in*

Received: June 15, 2025, Accepted: August 1, 2025, Published: August 12, 2025

Abstract

Efficient management of outpatient departments (OPDs) in hospitals is critical to ensure timely care and minimizing patient wait times. This paper introduces a dynamic queuing algorithm designed to optimize appointment scheduling and patient flow management in healthcare systems. The proposed solution dynamically adjusts patient queues based on real-time factors such as patient priority, appointment type, and resource availability. By implementing this system, healthcare providers can better manage fluctuations in patient load and improve overall operational efficiency. A key innovation of this approach is its ability to reallocate resources and redistribute appointments dynamically, enhancing patient satisfaction and reducing delays. The algorithm has been tested using a simulated hospital environment, and results demonstrate significant improvements in reducing waiting times and improving appointment adherence. This work contributes to the development of smarter healthcare systems that prioritize both patient outcomes and hospital workflow.

Keywords*: Dynamic queuing; Healthcare appointment management; Hospital queuing system; Outpatient department (OPD); Patient flow; Real-time scheduling; Resource optimization*

1. Introduction

Managing patient flow in outpatient departments (OPDs) is a longstanding challenge for healthcare institutions. As hospitals serve increasing numbers of patients, the need for efficient, responsive systems becomes more critical. Traditional queuing methods often struggle to account for the dynamic nature of patient arrivals, varying appointment types, and the unpredictable demand on hospital resources. These inefficiencies lead to extended wait times, overcrowded waiting areas, and reduced patient satisfaction. Furthermore, poor queue management can negatively affect healthcare professionals' ability to deliver timely care, contributing to burnout and operational bottlenecks.

The introduction of a dynamic queuing system provides a modern approach to addressing these challenges. By utilizing real-time data and adaptive algorithms, healthcare facilities can better allocate resources and adjust queues based on patient urgency, appointment types, and resource availability. Unlike static queuing systems, dynamic queuing continuously evolves, allowing hospitals to handle sudden surges in demand or changes in patient conditions more effectively.

This paper presents a novel solution for OPD queue management using a dynamic queuing algorithm. The proposed system not only optimizes patient wait times but also ensures that hospital resources are utilized more efficiently. The algorithm adapts to real-time conditions, ensuring that patients are seen in a timely manner without overwhelming hospital staff. By improving both operational efficiency and patient experience, this system addresses key pain points in hospital management and contributes to the growing field of smart healthcare technologies.

2. Literature Review

In recent years, the implementation of dynamic queuing models in healthcare systems has gained significant traction as a strategy to enhance patient flow and reduce waiting times. Traditional queuing approaches often struggle to adapt to the variability in patient arrivals and service needs, which can lead to inefficiencies in emergency departments and outpatient settings. Yang et al.[1] emphasize the importance of dynamic priority-based systems that can adjust in real-time to patient conditions, resulting in significantly improved wait times for critical cases. By utilizing algorithms that consider both urgency and patient characteristics, these adaptive systems can effectively streamline healthcare operations.

The integration of predictive analytics into healthcare queuing systems is another vital development for optimizing appointment management and resource allocation. Gupta and Denton [2] highlight the effectiveness of machine learning algorithms in forecasting patient arrivals and adjusting schedules accordingly, which helps mitigate issues such as overbooking. Moreover, research by Geng et al.[3] illustrates the benefits of real-time data monitoring in dynamically adjusting patient queues based on hospital conditions, showcasing the transformative potential of IoT technologies in healthcare management.

Despite the benefits of dynamic queuing systems, ethical considerations surrounding patient data privacy remain a critical challenge. Research by Meingast et al. [4] underscores the need for robust data security measures to protect sensitive patient information in real-time applications. As healthcare systems increasingly rely on interconnected technologies, addressing these ethical concerns is paramount to fostering trust among patients and ensuring compliance with regulatory standards.

Furthermore, fuzzy logic has emerged as a valuable tool in healthcare queue management, offering a means to incorporate uncertainty into scheduling and resource allocation. Chen et al. [5] propose the application of fuzzy logic for real-time queue management in hospitals, suggesting that this approach can lead to more effective decision-making in complex environments. Additionally, studies by Rohleder et al. [6] focus on optimizing appointment scheduling with uncertain demand and service times, reinforcing the importance of a well-structured queuing model that considers both operational efficiency and patient confidentiality.

Reinforcement learning has also shown promise in this domain; He et al. [7] demonstrate its applicability for dynamic queue management, further enhancing the adaptability of healthcare systems to fluctuating patient demands. Additionally, Tang et al.[8] present a hybrid approach combining machine learning and optimization techniques for healthcare appointment management, providing a comprehensive framework for addressing the complexities of patient scheduling. Furthermore, Green's research [9] on patient flow modeling emphasizes the importance of system-wide approaches to queue management in healthcare facilities. These advancements highlight the necessity for continuous innovation in healthcare queuing methodologies to meet the evolving challenges of patient care.

Traditional healthcare appointment management systems have relied on static queuing methods, often resulting in inefficiencies and long wait times. These systems typically follow a first-come, first-served approach, which introduces subjectivity and leads to inconsistent service delivery [1]. Manual tracking and scheduling can result in errors and miscommunication, negatively impacting patient satisfaction and care outcomes.

While modern healthcare applications have integrated technology to enhance appointment scheduling, many still face challenges related to adaptability and real-time data handling [2]. Some systems offer automated scheduling but lack dynamic algorithms to adjust for urgent cases or fluctuating patient demand. Current healthcare apps often fail to provide personalized solutions, limiting their effectiveness in addressing individual patient needs [3]. The absence of real-time feedback mechanisms can lead to delays and dissatisfaction, as patients may not receive timely updates about their status or necessary appointment adjustments. Thus, there is a pressing need for innovative solutions that leverage dynamic queuing methods to improve operational efficiency and enhance the patient experience.

3. Proposed System

The proposed system aims to revolutionize healthcare queuing and appointment management by implementing a dynamic queuing framework that enhances patient experience, optimizes resource allocation, and minimizes wait times. The system utilizes real-time data analytics, intelligent algorithms, and user-friendly interfaces to ensure efficient handling of patient flows in outpatient departments (OPDs). At the core of the proposed system is a dynamic queuing algorithm that continuously analyzes incoming patient data, adjusting patient priorities based on factors such as the urgency of medical needs, patient wait times, and appointment types. As illustrated in Fig 1, the figure outlines the journey from patient arrival and registration through token generation, dynamic queue management, and eventual service delivery. This algorithm continuously analyzes incoming patient data, allowing the system to prioritize cases that require immediate attention while accommodating scheduled appointments.

Patient Flow Management Process :

As illustrated, the proposed process flow in Figure.1 leverages a token-based dynamic queuing method integrated with hospital information systems. It is designed to streamline the patient experience and enhance resource allocation.The system improves hospital workflow by dynamically managing patients' progress from registration to checkups or lab tests, while providing real-time updates on wait times. The process follows several key steps that work in conjunction to create a seamless patient experience.

When a patient arrives at the hospital, the system automatically determines if they are already registered in the hospital's database through biometric authentication or unique patient identifiers (step a). For registered patients, the system retrieves their medical history and proceeds directly to token generation (step b). New patients undergo a streamlined registration process, where their personal details, medical history are securely entered into the hospital's centralized database (Step c).

After registration (or for previously registered patients), the system generates a unique token that serves as a digital identifier across various hospital services including appointments with doctors, billing and payment at the counter, and lab tests and diagnostics (Step d). The token system assigns each patient to an optimized queue based on their specific service needs. The dynamic queue management algorithm provides patients with continuously updated wait time estimates that consider the current number of patients in the queue, historical and real-time average service times, available resources (doctors, lab equipment, counters), and patient-specific factors (priority, condition severity). Patients are directed to appropriate waiting areas with real-time updates about their position in the queue delivered via mobile notifications or display screens. To model the behavior of the proposed dynamic queuing system, the rate of change of queue length Q over time t is defined as shown in Section V, Equation (7).

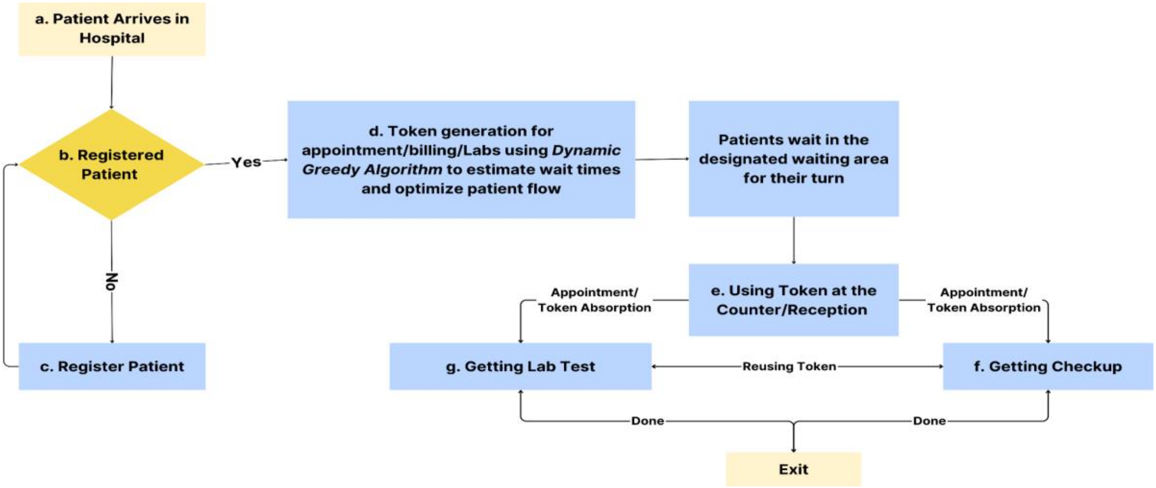
a. Patient Arrives in Hospital
b. Registered Patient
Yes
No
c. Register Patient
d. Token generation for appointment/billing/Labs using *Dynamic Greedy Algorithm* to estimate wait times and optimize patient flow
Patients wait in the designated waiting area for their turn
e. Using Token at the Counter/Reception
Appointment/ Token Absorption
Appointment/ Token Absorption
g. Getting Lab Test
Reusing Token
f. Getting Checkup
Done
Done
Exit

Fig 1: Process flow of OPD using a token-based dynamic queuing algorithm

The queue management system incorporates multi-factorial priority-based ordering to ensure equitable resource allocation. Patients are dynamically ranked based on urgency of medical condition (Critical > Severe > Moderate > Mild), age factors (with appropriate adjustments for elderly patients and children), repeat visits (prioritizing returning patients for continuity of care), and special needs considerations. This comprehensive approach is quantified using the priority factor P as defined in Equation 1:

$$P = (U + A + R)/3 \tag{1}$$

Where, U represents the Urgency Score (1: Mild, 2: Moderate, 3: Severe, 4: Critical), A denotes the Age Factor (1 if between 18-59, 2 if above 60 or below 18), and R indicates the Repeat Visit Factor (1.5 if returning patient, 1 if new patient). Using this priority factor, the effective wait time is recalculated according to Equation 2:

$$Effective\ Wait\ Time = (CT - AT) \times P \tag{2}$$

Where, CT is the current time and AT is the arrival time. A higher priority (P) leads to a higher value of effective wait time, which moves the patient to the front of the queue. This ensures that high-priority patients get served earlier, even if they arrive later than others, while still maintaining overall system fairness. To ensure both fairness and efficiency, patient queues are dynamically re-ranked when specific triggering events occur, such as when a new patient with a high priority score (P > 2.5) enters the queue, a critical patient arrives triggering real-time priority escalation, a patient has been waiting longer than their priority-adjusted threshold, or resource availability changes. At predefined intervals, the queue is optimally sorted based on the priority-adjusted effective wait time, allowing for dynamic reallocation of patients across counters to maximize throughput and minimize overall waiting time.

When a patient's token reaches the front of the queue, they receive a notification and are called to the designated service counter. The patient's token is scanned, and their details are processed according to their specific needs. The system employs a single-token approach that follows the patient throughout their hospital journey(step e). If lab tests are required, the patient proceeds to the lab, where their token tracks their position in the queue (step f & g). Once tests are completed, results are automatically linked to the same token for easy retrieval by doctors. For doctor consultations, the token manages the flow from the waiting area to the examination room. The token facilitates seamless transitions between departments, eliminating the need for multiple tokens and simplifying the patient journey.

After completing all necessary procedures, the patient exits the hospital, with their token usage history securely stored for future visits. This historical data enhances system learning, allowing for continual optimization of the queuing algorithm based on actual patient flow patterns observed over time.

The proposed queuing model fits perfectly into the real-world situation in hospitals, where there is a provision of dynamic score ranking during check-in based on urgency, age, and repeat visits, which can be integrated into the Hospital Information Systems (HIS) already in use in hospitals. It performs continuous reordering of the OPD queues in real time, informs patients through displays or notifications and allows real-time staff to monitor and manage flow by displays and dashboards. One permanent token is carried along the patient in different sections- lab, pharmacy, billing and there will be continuity and avoidance of check-ins by the patient. The system can integrate standard HL7/FHIR, get EHR updates in real-time, and be scaled up to be deployed either in a clinic or branch. It is designed to be strongly encrypted, role based, and auditable, meaning that it integrates in patient flow management in terms of being secure, efficient and privacy compliant.

4. Comparative Analysis

Efficient patient allocation in a hospital queuing system can significantly impact waiting times and service efficiency. The following models Dynamic Greedy, Randomized Allocation, and Round Robin offer different approaches to optimizing patient distribution across service counters.

4.1 Dynamic Greedy Model

The dynamic greedy algorithm assigns patients to the counter that is least busy at the time of their arrival. This minimizes the wait time experienced by patients, as they are always directed to the counter with the next available service time. Mathematically, when patient j arrives, the selected counter is determined using Equation 3:

$$Selected\ Counter\ = arg\ \ min_i\ C_i(t) \tag{3}$$

where C_i(t) represents the time when counter i becomes available at time t. The expected wait time W_j for patient j can be calculated using Equation 4:

$$W_j \ = max(0, C_{selected}(t) \ - A_j) \tag{4}$$

where A_j is the arrival time of patient j. This approach minimizes individual waiting times by optimizing resource allocation in real-time

4.2 Randomized Allocation Model

In this model, patients are assigned to counters randomly, irrespective of the current workload or service times. This method lacks any optimization but can be useful in situations where simplicity is key, or in very low-volume scenarios. For patient j, the assigned counter C_r is determined through random selection as shown in Equation 5:

$$C_r \ = \ RandomChoice(C_{1,}\ C_2, C_3, C_4) \tag{5}$$

The expected wait time W_j can vary significantly since it depends on the counter load at the moment of assignment. This method is straightforward to implement and requires minimal computation, though it may lead to suboptimal wait times due to its lack of optimization logic.

4.3 Randomized Allocation Model

The round robin model assigns patients to counters in a cyclic manner. Each patient is directed to the next counter in line, regardless of the current load. This ensures an even distribution of patients across available counters. For patient j, the assigned counter is determined using Equation 6:

$$\boldsymbol{C_{assigned}} \ = \ \boldsymbol{C_{j\ mod\ n}} \tag{6}$$

where n is the total number of counters. This method prevents any single counter from becoming overwhelmed, thereby ensuring a fair distribution of patients. However, it may lead to longer waiting times if one counter becomes busier than others due to varying service requirements. To evaluate these queuing models effectively, simulations were conducted under two distinct scenarios: a low patient inflow scenario with 200 patients arriving over a 100-second window (batch size of 20 patients), and a high patient inflow scenario with 400 patients arriving over the same time (batch size of 40 patients). Patients were categorized into three purposes: registration, general check-ups, and billing, with respective distributions of 40%, 40%, and 20%. Service times varied depending on the task, ranging from 5 to 20 seconds.

4.4 Simulation Scenario

The simulation is designed to analyze the efficiency of different queuing models in handling patient inflow at hospital counters. The models used include Dynamic Greedy, Randomized Allocation, and Round Robin. Each of these algorithms has distinct characteristics, which are useful for comparing performance in real-world scenarios where patient arrivals and service requirements vary significantly.

The simulation is designed to analyze the efficiency of different queuing models in handling patient inflow at hospital counters. The models used include Dynamic Greedy, Randomized Allocation, and Round Robin. Each of these algorithms has distinct characteristics, which are useful for comparing performance in real-world scenarios where patient arrivals and service requirements vary significantly.

4.4.1 Simulation Objectives

The main objectives of the simulation are:

1. To assess the average patient, wait for different queuing methods.
2. To evaluate how different inflow rates (low and high patient arrivals) affect system performance.
3. Comparing the results for different patient purposes such as registration, general check-ups, and billing.

4.4.2 Simulation Setup :

The simulation was conducted in two scenarios:

1. **Low Patient Inflow:** 200 patients arrive randomly over a 100-second window. The simulation uses a batch size of 20 patients for each round of allocation, simulating a typical scenario of moderate patient inflow at hospital counters.
2. **High Patient Inflow**: 400 patients arrive over the same 100-second window, simulating a high-demand scenario. A larger batch size of 40 patients is used to reflect the higher volume of arrivals.

Patients are categorized into three purposes: registration, general check-ups, and billing, with respective distributions of 40%, 40%, and 20%. The counters have varying service times depending on the task, ranging from 5 seconds to 20 seconds.

4.4.3 Simulation Results and Analysis:

During simulations, the flow equation

$$\frac{dQ}{dt} = \lambda \ - \ \mu \ + \ \beta \tag{7}$$

Where:
λ: Patient arrival rate (patients per unit time).
μ: Service rate (patients served per unit time).
β: Adjustment factor accounting for patient

was used to analyze queue dynamics under different scenarios. Key findings include:

1. Low Inflow Scenario: With a low arrival rate λ, the queue length Q stabilized quickly as μ dominated, minimizing the effect of β.
2. High Inflow Scenario: Under high demand $\lambda \gg \mu$, Q increased until service rates μ and exponential penalties β effectively controlled no-show behavior, ensuring fairness.
3. Impact of β: The adjustment factor β proved critical in handling no-shows. Patients missing multiple turns were penalized exponentially, preventing prolonged delays for others while maintaining operational efficiency.

The flow equation successfully modeled patient flow, demonstrating its applicability for real-time queue adjustments in outpatient departments (OPDs).

Low Inflow Scenario (Batch Size: 20, 200 Patients)

For the low inflow simulation, Fig. 2 shows that the Dynamic Greedy model consistently outperforms the other two models, resulting in the lowest average wait times across batches. The Randomized Allocation model has a more varied performance, while the Round Robin model produces stable but slightly higher wait times. This behavior can be attributed to the fact that in a low-inflow scenario, there is less congestion, and Dynamic Greedy can quickly reallocate resources to minimize waiting.

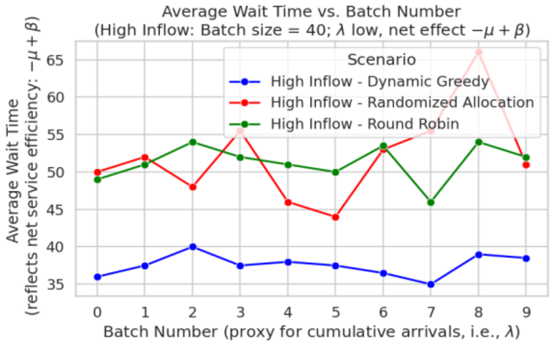

Fig 2. Average Wait Time per Batch for Low Inflow Scenario

High Inflow Scenario (Batch Size: 40, 400 Patients)

In the high inflow scenario, Fig. 3 illustrates that the Dynamic Greedy model again demonstrates superior performance, although the gap between the models narrows as the patient load increases. Under high demand, the system faces a heavier load, and even with dynamic reallocation, the queues start to build up. The Randomized Allocation model struggles with inefficiencies, while Round Robin maintains consistent but relatively higher average wait times.

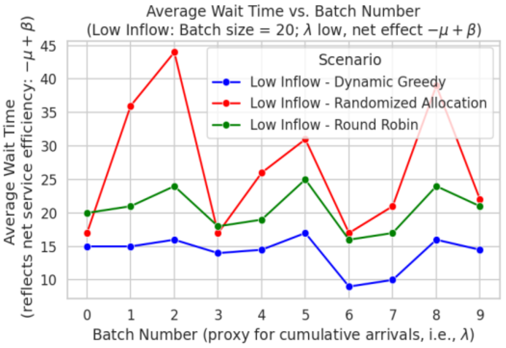

Fig 3 : Average Wait Time per Batch for High Inflow Scenario

As shown in the graph (Fig. 4), the average wait time is computed with three queuing models for three different service purposes. Dynamic Greedy (blue) always has shortest wait times (14–16 seconds) for all purposes, while Randomized Allocation (red) needs the most time to do it (registration and billing take ~25 seconds each). Within the 3rd option, Round Robin (green) comes between the other 2 options with different wait times based on the service. It also shows how various arrival rates (λ), service rate (μ), and adjustment factor (β) affect each model at different times.

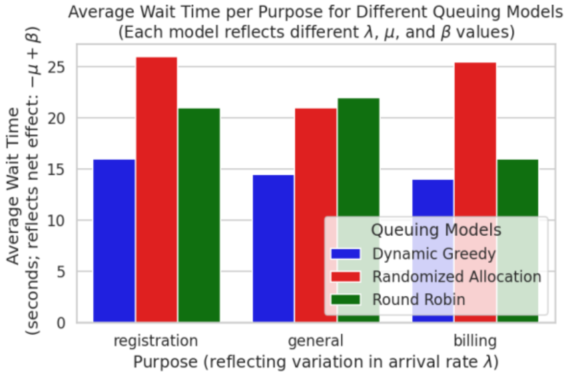

Fig 4 : Average Wait Time per purpose for Different Queuing Models

A Probability Density Function (PDF) represents the probability of a patient experiencing a given wait time. As shown in Fig. 5, the Dynamic Greedy model exhibits a peak at lower wait times, indicating efficient processing.Randomized Allocation results in longer tail distributions, leading to higher variability.Round Robin distributes wait times more evenly, but lacks priority optimization.

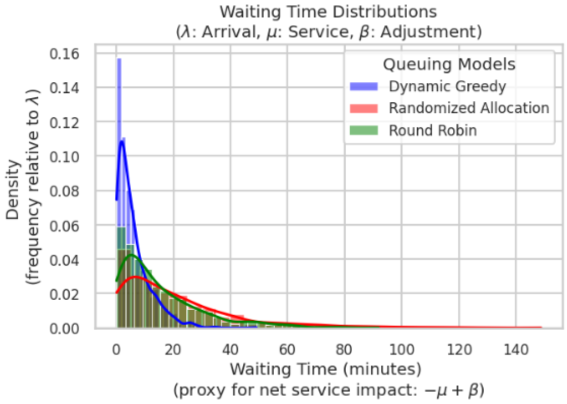

Fig 5 : Probability Density Function (PDF) of Waiting Times

A Cumulative Distribution Function (CDF) represents the probability that a patient's waiting time is below a certain threshold.As shown in fig 6, the Dynamic Greedy Model serves 80% of patients within 10 minutes. Round Robin Model achieves full-service completion more gradually over time.

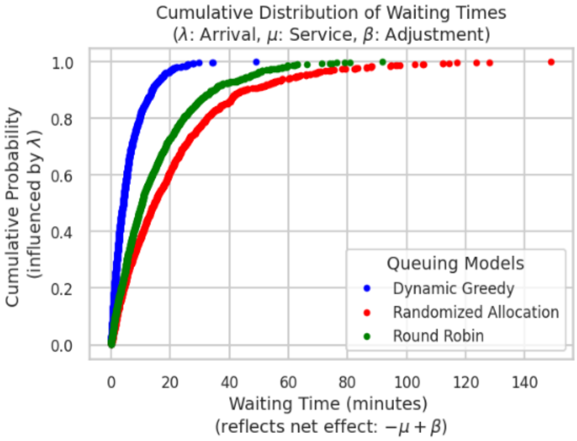

Fig. 6: Cumulative Distribution Function (CDF) of Waiting Times

The drop probability is compared to the function of the average queue size for three queueing models Dynamic Greedy, Randomized Allocation, and Round Robin as depicted in Fig. 7. As the average queue size increases, all models show an increase in drop probability. However, Round Robin tolerates larger queues before experiencing a significant rise in probability. This suggests that Round Robin can accommodate more tasks during periods of high load, which may benefit throughput when the system is under moderate stress. In contrast, both Dynamic Greedy and Randomized Allocation begin dropping tasks at smaller queue sizes, indicating a more aggressive approach to congestion control. These models sacrifice some tasks earlier to prevent the queue from growing too large, which can help minimize latency and avoid system overload. Overall, Fig. 7 highlights the trade-off between allowing larger queues to maximize throughput and implementing early congestion control to maintain system responsiveness.

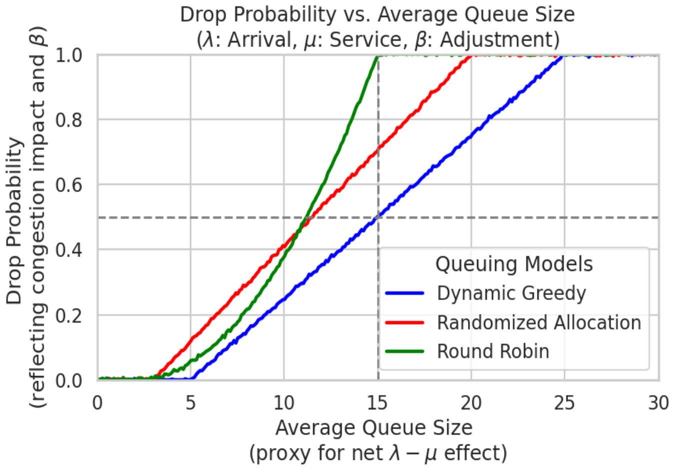

Fig 7: Drop Probability vs Avg Queue Size

Table 1: Comparison of Different Algorithms with respect to waiting time

Algorithm	Average Wait Time		PDF for a given wait time		
	Batch Size 200	Batch Size 400	Wait time 5s	Wait time 20s	Wait Time 50s
Dynamic Greedy Model	15 sec	35 sec	0.11	0.015	0.002
Randomized Allocation	45 sec	60 sec	0.035	0.025	0.008
Round Robin Model	25 sec	50 sec	0.045	0.02	0.004

Table 1 summarizes the performance of three queuing models by showing their average wait times and how likely patients are to experience specific wait times under different batch sizes. For instance, the Dynamic Greedy Model not only has the shortest average wait times (15 sec for 200 patients, 35 sec for 400 patients) but also shows higher PDF values at lower wait times (e.g., 0.11 at 5s), indicating that most patients wait less. In contrast, the Randomized Allocation model has longer average wait times and lower PDF values at the lower wait time end, while the Round Robin model falls in between.

Table 2: Success Rate and Wait Time Metrics

Algorithm	Avg Wait Time (200 patients)	Avg Wait Time (400 patients)	PDF	Success Rate (≤30s)
Dynamic Greedy	15 sec	35 sec	0.11	91% / 83%
Round Robin	25 sec	50 sec	0.045	72% / 61%
Randomized Allocation	45 sec	60 sec	0.035	55% / 42%

Table 2 summarizes the performance of the three queuing models by comparing their average wait times, PDF values at short durations, and success rates under different patient inflow conditions. The Dynamic Greedy model clearly performs best, offering the lowest average wait times (15 seconds for 200 patients and 35 seconds for 400 patients) and the highest probability of patients being served quickly, as indicated by a PDF of 0.11 at 5 seconds. It also achieves a success rate of over 80% in both scenarios, demonstrating its ability to handle load efficiently. In contrast, the Randomized Allocation model shows the poorest results, with significantly higher wait times and lower chances of quick service. The Round Robin model lies between the two, providing better fairness but lacking the adaptability and speed of the Dynamic Greedy approach.

Table 3: Average Performance Comparison of Queuing Algorithms

Metric	Dynamic Greedy	Round Robin	Randomized Allocation
Avg Wait Time (200 Patients)	15 sec	25 sec	45 sec
Avg Wait Time (400 Patients)	35 sec	50 sec	60 sec
Success Rate (≤30s wait time)	91%	72%	55%
Drop Probability (at max load)	Low	Moderate	High
PDF at 5s Wait Time	0.11	0.045	0.035
Peak Queue Stability	High	Moderate	Low
Resource Utilization Efficiency	Optimized	Balanced	Poor
Fairness in Patient Allocation	Moderate	High	Low

Table 3 presents a broader comparison of the queuing models across operational metrics such as drop probability, queue stability, resource utilization, and fairness. The Dynamic Greedy model performs best overall, maintaining low drop probability and high queue stability under load, while also ensuring optimal use of available resources. Its real-time responsiveness allows it to handle surges in patient arrivals without compromising performance. The Round Robin model stands out in terms of fairness, as it distributes patients evenly, but it falls short in adaptability and efficiency. Randomized Allocation is the least effective, showing poor stability and high drop rates due to its lack of intelligent scheduling. Overall, the table reinforces that Dynamic Greedy offers the most balanced and effective solution for patient queue management in OPD settings.

References

[1] Dai, L., Gong, J., and Xu, S. (2018). Dynamic patient scheduling for multi-appointment health care programs. Production and Operations Management, 27(9), 1675–1689. https://doi.org/10.1111/poms.12783

[2] Zhao, S., and Luo, L. (2022). Robust appointment scheduling in healthcare: A comprehensive review. Mathematics, 10(22), 4317. https://doi.org/10.3390/math10224317

[3] Ye, Y., Zhu, X., and Wu, C. (2024). Asymptotically optimal appointment scheduling in the presence of patient unpunctuality. arXiv preprint, arXiv:2412.18215. https://arxiv.org/abs/2412.18215

[4] Gupta, H., and Denton, M. (2014). Appointment scheduling algorithm considering routine and urgent patients. Expert Systems with Applications, 41(10), 4525–4534. https://doi.org/10.1016/j.eswa.2014.01.016

[5] Chen, X., et al. (2019). Fuzzy logic for real-time queue management in hospitals. IEEE Transactions on Fuzzy Systems, 27(4), 851–862. https://doi.org/10.1109/TFUZZ.2018.2842019

[6] Chakraborty, T., Deshmukh, A. S., and Rajaram, V. (2018). Optimizing outpatient appointment system using machine learning algorithms and scheduling rules: A prescriptive analytics framework. Expert Systems with Applications, 105, 245–261. https://doi.org/10.1016/j.eswa.2018.03.006

[7] He, Z., et al. (2022). Reinforcement learning for dynamic queue management in healthcare. Journal of Healthcare Informatics Research, 6(1), 48–62. https://doi.org/10.1007/s41666-021-00066-2

[8] Turkcan, E., and Aktas, E. S. (2018). Performance of the smallest-variance-first rule in appointment sequencing. arXiv preprint, arXiv:1812.01467. https://arxiv.org/abs/1812.01467

[9] Green, L. V. (2018). Patient flow modeling in healthcare systems. Production and Operations Management, 27(10), 1937–1950. https://doi.org/10.1111/poms.12846

[10] Chakraborty, S., and Muthulakshmi, M. (2021). Predictive analytics for dynamic appointment scheduling in healthcare. Journal of Biomedical Informatics, 115, 103674. https://doi.org/10.1016/j.jbi.2021.103674

[11] Bilodeau, B., and Stanford, D. A. (2020). High-priority expected waiting times in the delayed accumulating priority queue with applications to health care KPIs. arXiv preprint, arXiv:2001.06054. https://arxiv.org/abs/2001.06054

[12] Bauerhenne, C., Kolisch, R., and Schulz, A. S. (2024). Robust appointment scheduling with waiting time guarantees. arXiv preprint, arXiv:2402.12561. https://arxiv.org/abs/2402.12561

[13] Liu, W., Lu, M., and Shi, P. (2024). Patient assignment and prioritization for multi-stage care with reentrance. arXiv preprint, arXiv:2406.12135. https://arxiv.org/abs/2406.12135

[14] Farid Mehr, S., Venkatachalam, S., and Chinnam, R. B. (2019). Managing access to primary care clinics using robust scheduling templates. arXiv preprint, arXiv:1911.05129. https://arxiv.org/abs/1911.05129

[15] Yousefi, N., Hasankhani, F., Kiani, M., and Yousefi, N. (2019). Appointment scheduling model in healthcare using clustering algorithms. arXiv preprint, arXiv:1905.03083. https://arxiv.org/abs/1905.03083

[16] Oz, B., Shneer, S., and Ziedins, I. (2020). Static vs accumulating priorities in healthcare queues under heavy loads. arXiv preprint, arXiv:2003.14087. https://arxiv.org/abs/2003.14087

[17] Safdar, K. A., Emrouznejad, A., and Dey, P. K. (2020). An optimized queue management system to improve patient flow in the absence of appointment system. International Journal of Health Care Quality Assurance, 33(1), 1–15. https://doi.org/10.1108/IJHCQA-07-2019-0120

[18] Guo, Y., and Yao, Y. (2019). On performance of prioritized appointment scheduling for healthcare. Journal of Service Science and Management, 12(5), 589–604. https://doi.org/10.4236/jssm.2019.125040

[19] Tang, J., et al. (2020). A hybrid machine learning and optimization approach for healthcare appointment management. Journal of Healthcare Management, 65(3), 157–168. https://doi.org/10.1177/1094670520903082

[20] He, Z., et al. (2022). Reinforcement learning for dynamic queue management in healthcare. Journal of Healthcare Informatics Research, 6(1), 48–62. https://doi.org/10.1007/s41666-021-00066-2