Novel Algorithms to Manage Overcrowded Emergency Departments with Varying Levels of Congestion: Prioritize Patients under the Dilemma of Efficiency, Satisfaction, and Fairness

Completed for the Certificate in Scientific Computation and Data Sciences
Spring 2023

Vivian Bui
Certificate in Scientific Computation and Data Sciences
Department of Statistics and Data Sciences
College of Natural Sciences

Arya Farahi, Assistant Professor
Department of Statistics and Data Science

1 Introduction

Emergency healthcare plays a pivotal role in responding to acute injuries and immediate, life-threatening trauma. In situations where logistical and financial barriers prevent access to preventive or long-term care, emergency healthcare serves as a safety net that provides diagnostic, evaluation, and treatment for patients who often present to care at the point of systemic to severe illness. According to the National Center for Health Statistics (NCHS), the emergency department visit within the twelve-month period of the study for adults aged 18 and over in the United States from 2019 to 2022 tops at 23 percent on average [1]. Another NCHS report published in 2021 revealed that, excluding infectious diseases and catastrophic healthcare events such as COVID-19, the overall emergency department visit rate on average is 40 visits per 100 persons across all age groups for the past five years of the research [2]. As vital the role of emergency healthcare to the daily life of Americans is, its impact nonetheless extends beyond individual health to affect "the overall health and productivity of the population" [3]. By boosting the well-being of individuals and assuring Americans the care and safety when needed regardless of their prolonged or situational hardships, emergency care produces and reinstates the capability of people to live, work, and further contribute to becoming productive members of society. Well-functioning emergency departments with effective and efficient operations, therefore, serve as indicators of invested public policy and a thriving local economy.

The essential and accessible nature of emergency departments (ED), however, leads to overcrowding and causes significant challenges in operations, management, and patient satisfaction. As defined by the American College of Emergency Physicians, overcrowding is "a situation that occurs when the identified need for emergency services exceeds available resources for patient care in ED, hospital, or both" [4]. Such problem resulted in long boarding times, "medical errors," "compromises to patient privacy," and "increased mortality" that put strain on the medical systems [5]. To make it more severe, overcrowding not only lengthen the wait time for patients but further forces patients to leave without being seen (LWBS) by a medical professional. As discovered by Yvonne Ng and Stuart Lewena (2011), ED overcrowd is the dominant factor underlying LWBS problems and LWBS subsequently reflects the severity of ED overcrowding situation [6]. Not receiving the necessary diagnosis and treatment could leave these LWBS patients in suffering "deleterious consequences including death and disability" [7]. Furthermore, from an economic standpoint, "LWBS also constitutes lost ED revenue and potentially lost hospital revenue if the patient's condition would warrant further inpatient admission"[7].

Rigorous research efforts have been made to tackle the overcrowding issues in ED including proposing new triage procedure [8], expanding alternative resources [9], downsizing beds [10], restructuring [11], ambulance diversion [12], and discrete-simulation with heuristic optimization [13]. However, challenges remain in finding strategies that are adaptive and volatile to the perpetual changes in the emergency departments, especially during the situational surge in patient influx. To date, no prior research has provided a definitive approach for optimizing the selection of patients in the waiting queue.

In an effort to address the challenges posed by overcrowding in emergency departments, we present a simulation model of an emergency room (ER) developed using Python programming language. Our simulation model emulates various scenarios of patients waiting for treatment under varying degrees of overcrowding, which are pre-configured prior to the execution of the program. Central to the simulation is the algorithm used to select patients from the waiting queue. We introduce eight different algorithms to select patients. Through the simulation of multiple scenarios, we investigate the performance of various queue selection algorithms under different levels of overcrowding. We provide metrics and visualization to observe, analyze, and discuss simulated results.

In the subsequent section of this paper, we present research conducted in the field of addressing overcrowding issues in emergency healthcare systems. Building on this review, we introduce the problem setup and provide an in-depth description of the simulator used in our study. This includes a detailed discussion of the module architecture, the patient and ER-level operations, as well as the initial conditions required to execute the simulation. Furthermore, we provide a comprehensive overview of the selection algorithms and evaluation metrics used in our study, highlighting the key features of each approach. Subsequently, we proceed to analyze and discuss the results obtained from our simulation experiments. This analysis involves a detailed examination of the performance of various queue selection algorithms under different levels of overcrowding and provides insights into the effectiveness of each approach in mitigating the impact of overcrowding in emergency healthcare systems.

2 Literature Review

Factors contribute to overcrowding in the emergency department can be classified into three categories: input, throughput, and output factors [14]. Input factors include the waiting time, the number of patients visiting the ED, as well as their severity and complexity (or triage score). Throughput, or internal factors, are those captured by the processing time such as the time to diagnose, give treatment, discharge, hospitalize, and transfer. Finally, output factors encompass the patients boarding in the ED, the availability of hospital beds, and the delay of transport to leave the ED [14]. The targeting subject for alleviating the situation of overcrowding in ED, hence, centralizes around the three factors mentioned above.

The need to improve the efficiency of ED without sacrificing the quality of care naturally leads to the widespread application of operations research and operations management techniques (OR/OM) in the field [15]. The OR/OM techniques used to describe and analyze patient flow in EDs are classified into two categories: prescriptive and descriptive techniques [16]. Descriptive techniques can be further categorized into analytical modeling, simulation modeling, and statistical modeling [17]. In the context of EDs, "queuing theory is the most prevalent analytical modeling technique" [18]. However, "as analytical models mostly rely on closed-form mathematical formulations, they are not suitable to model the complex, stochastic and dynamic nature of healthcare system without introducing simplifying assumptions" [18]. Statistical modeling, on the other hand, is entirely based on empirical data to estimate system performance and the relationship between system parameters and performance measures [17, 19]. When no data is available or when the study tends to provide a general framework than system-specific, statistical modeling poses difficulty for execution. Simulation, therefore, is an optimal choice for our study. The advantages of the simulation lie in its capability to operate in a time-dependent environment and replicate the complex patient influx by approximating real-life behavior.

In 2016, Nahhas et al. proposed a discrete-event simulation approach to investigate the impact of various throughput factors, such as bed availability and staffing, on crowding in healthcare facilities [13]. The focus of the study was about cost reduction by lowering the number of patients who left without being seen (LWBS). To conduct the experiments, five scenarios were designed and implemented. The first scenario was set as the basic scenario, which utilized the initial conditions of the simulator without any modification. The data used in this scenario was provided by SIMIO LCC, a private company that organized the simulation competition.

In the second, third, and fourth scenarios, the authors investigated the effects of modifying operations in procedural rooms, changing the division of tasks among staff members, and increasing the number of hours worked by all staff members per shift, respectively. The results from each scenario were integrated into subsequent scenarios to determine the combined effect of all modifications.

In the final scenario, all previous modifications were combined with further adjustments to improve the precision, flexibility, and sophistication of shift patterns. To generate solutions and increase the quality of results, the authors used the technique of heuristic optimization and imposed upper and lower bounds to scale down the number of possible configurations. Through the accumulative design of the simulator, the researchers found that adjusting through puts in the second and third scenarios resulted in the most optimal outcomes in the fourth and final scenarios, according to their pre-defined cost reduction metrics [13].

Nahhas et al.'s simulator presents several noteworthy features that contribute to its robustness. Firstly, the simulator incorporates the relationship between the availability of staffing and infrastructure resources with respect to waiting time, thereby capturing a vital aspect of healthcare systems [13]. Secondly, the simulator takes into account the acuity level of patients when selecting them, thereby providing a more realistic representation of patient flow in healthcare facilities. Additionally, the simulator features a distinct attribute for the LWBS metric assigned to each patient, which provides a diversified range of patient characteristics and event occurrences in the simulation environment. This design element enables more granular analysis of the impacts of LWBS on system performance and efficiency. One major disadvantage of Nahhas's simulation, however, is the reliance on the staffing resources. In the context of severe shortage of healthcare workers, the simulator is more prone to errors and impracticality.

Before Nahhas et al., another complex discrete model with space syntax analysis was created by Morgareidge et al. in 2014. [11] Their focus was on redesigning the complete structure of ED and its impact on the possible impact of such reorganization. However, streamlining the entire inputs, throughputs, and outputs proved to pose further challenges, especially in the uncertainty of whether the architectural changes could solve the inherent problems caused by harder-to-control factors such as patient inflow and their severity. Another approach that gained population between 2005 to 2015 was known as Ambulance Diversion (AD). The gist of AD is to reduce overcrowding in one ED by outsourcing patients to other EDs. Cooney et al. (2011) and Lin et al. (2015) defined a crowding index and generated discrete event simulation [12, 20]. The authors further proposed different AD policies to conclude that reallocation of patients positively alleviates overcrowding in one ED by equalizing the overall crowding level across EDs at the local. This practice, in fact, has been widely adopted by healthcare providers. The reliance on a peer network for support systems across EDs, however, requires a comprehensive communication channel and effective means of logistical navigation that are resilient to managerial strain and stress.

Other research on ED simulations dates back to the early 2000s, with many studies focused on optimizing throughput factors to reduce overcrowding. For example, Conelly and Bair (2004) proposed a new triage procedure to improve patient flow [8], while Kreke et al. (2004) suggested expanding alternative resources at an acceptable cost [9]. Wiinamaki and Dronzek (2003) proposed a hybrid approach that involved expanding certain aspects of the system while downsizing the number of beds to reduce costs in other areas [10].

However, to our knowledge, no previous research has explored the potential benefits of optimizing ED operations by selectively admitting patients who are waiting for care. This represents an important area for future investigation, as it may have significant implications for improving patient outcomes and reducing overcrowding in the ED.

3 Problem Setup

At the beginning of the simulation, the emergency room will be allocated a predetermined number of unoccupied beds for new patients. Every hour, a stochastic number of patients will arrive at the ER and require bed assignments. Upon arrival, each patient will undergo a triage process and will then be placed into a waiting queue. If a bed becomes available, a waiting patient who has been prioritized according to the selected algorithm will be assigned to the bed. Subsequently, after a duration of time has elapsed in the emergency bed, the treated patient will vacate the bed, making it available for the next waiting patient. The simulation will cease after running for the designated number of days as per the assignment.

Treatment as Bed Assignment. In our simulation, it is important to clarify that the assignment of an emergency bed encompasses three key components: (1) the receipt of necessary scanning procedures, (2) consultation with a doctor, and (3) receipt of treatment. We operate under the assumption that all incoming patients will require a bed, and we interchangeably utilize the terms "discharged" and "treated" to denote patients who have completed their stay in the ER. Accordingly, the term "discharge" includes any other potential outcomes that a patient may experience subsequent to their stay in the ER.

All Patients are Approximately Unique. All attributes pertaining to an incoming patient in our simulation are randomly generated, resulting in each patient being essentially unique. It should be noted that there may or may not be instances of "revisiting" patients, although the current version of the simulation does not account for such similarities in patient attributes.

Assumption of Triage. An acute level will be randomly assigned to a new patient as a way to simulate the result of triage without the need of detailing the entire triage process. Furthermore, we assume that since the first evaluation of triage, the patient's medical condition will remain unchanged during the waiting period until they are seen by a doctor. In other words, the patient's condition is assumed to neither improve nor deteriorate while waiting for medical attention which affects the triage result.

Calculation of Wait Time. Wait time calculation for a patient starts after triage has occurred and stops immediately at the time received a bed.

Left Without Being Seen as the Floor and Ceiling Function. The "Left Without Being Seen" (LWBS) feature in our simulation allows for customization by assuming that a patient will leave the emergency room if their wait time exceeds a predefined maximum benchmark. This feature has several advantages. Firstly, it reflects the reality that patients will not wait indefinitely for medical attention. Secondly, it reduces the overall wait time for the remaining patients in the queue. In terms of satisfaction scores, the LWBS feature serves to (1) eliminate negative scores and (2) potentially improve the satisfaction scores of the remaining patients. In other words, this feature sets a ceiling threshold for the maximum number of minutes a patient can wait, and establishes a minimum floor for satisfaction scores that cannot go lower than 0 per individual. [1]

Levels of Overcrowding. There is no universal or national metric to measure the level of crowding in emergency departments. The estimation of "High", "Medium", and "Low" level of overcrowding, in fact, is made in relation to the rate of LWBS patients. Jillian et al. (2021) found that ED crowding is the single most important factor that determined the likelihood and the percentage of LWBS patients in an emergency healthcare facility [21]. Hwang et al. (2011), in fact, proposed a metric that then became a widely-adopted practice in which LWBS is considered as one of the indicators for overcrowding [22]. With that being said, our determination for the levels of overcrowding is relative to the clinical data in the United States. As reported by Janke et al., "Median (IQR) hospital LWBS rates nearly doubled from 1.1 % (0.5%-2.5%) in 2017 to 2.1% (0.6%-4.6%) by the end of 2021. Among the worst performing hospitals at the 95th percentile, 10.0% of ED patients left before a medical evaluation at the end of 2021, compared with 4.4% in the 95th percentile in January 2020 and 4.3% at the beginning of 2017 [23]." As we are interested in observing how the algorithms behave in extreme overcrowding scenarios, we configure the initial conditions in a way that will result in the LWBS rate at least equal to or larger than the provided clinical data. The results are then divided into three scenarios, where the difference between levels justifies the smaller percentage of overall LWBS patients across algorithms.

4 Simulator

4.1 Modules Architecture

The core architecture consists of two essential components: classes and operations. The major classes, including Person, Patient, Queue, and Emergency Room, form the foundation of the architecture. The person serves as the parent class of the Patient, while the Emergency Room class efficiently stores all the instances of waiting and treating patients. On the other hand, the Queue class specifically manages patients who are waiting for bed assignments.

The operations within the architecture are categorized into two levels: patient-level and ER-level. PatientOps and Satisfaction modules are designed to operate at the patient level, providing essential functionalities related to patient management. Meanwhile, EROps and Select Queue modules are responsible for ER-level operations, optimizing the management of the Emergency Room and patient assignment to the Queue. The Main module brings together all the components of the program, and Main Sim is where the simulation is executed. Users have the flexibility to determine the number of simulations to run in the Main Sim. See the flow diagram of Figure 1.

4.1.1 Classes

1. Class: Person. The class "Person" includes attributes and methods that govern the ID of all individuals in the ER environment. The attributes associated with the "Person" class include "ID" and "Code". The "ID" attribute is an integer value that is assigned to each person based on the order in which they arrive at the ER. The ID number is initialized to 1 and is incremented by 1 for each subsequent person. On the other

[1] Refer to section 4.2.2 - Operations at Patient-level, Satisfaction to understand the meaning of the "0" flooring value in calculating satisfaction score

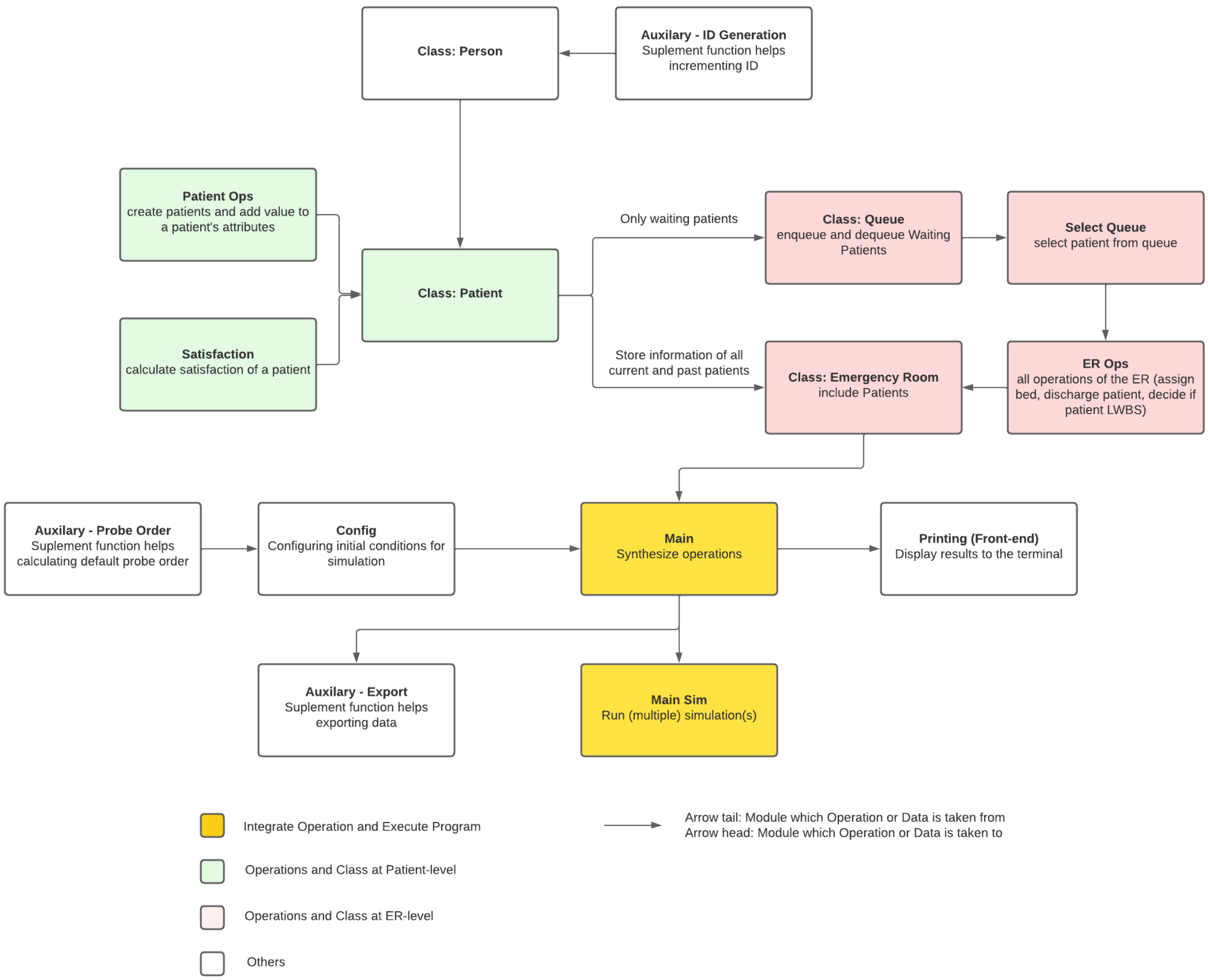

Figure 1: Modules Architecture

hand, the “Code” attribute is a string that represents a default code assigned to each patient, denoted as “P”.

In addition to the attributes, the “Person” class also includes methods, which are functions that define the behavior of the class. The methods associated with the “Person” class are “Get ID” and “Get code”. “Get ID” method is designed to retrieve the ID of a person, allowing access to the unique identification number assigned to them upon arrival at the ER. Similarly, “Get code” method enables retrieval of the code associated with a person, providing access to the default code “P” that is assigned to them as a patient.

Overall, the “Person” class serves as a foundational element of the ER system, encapsulating identification of individuals within the ER environment. For further information, refer to Class Person in Appendix Class Definition Table.

2. Class: Patient. The Patient class inherits attributes and methods from the Person class in the simulation environment of an emergency department (ER) system. The attributes of the Patient class include Age (int), Acute level (int), Status (from 0 to 2, default = 0), Pain level (int), Day coming (int), Time coming (time), Length stay in ER (int), Priority score (float), Day assigned (int), Time assigned (time), Day discharged (int), Time discharged (time), and Satisfaction score (float, default = 5.0).

The Age attribute represents the age of the patient and can range from 1 to 100 years old. The Acute level attribute represents the severity of the patient's condition and can have values from 1 to 5, with a lower value indicating a higher level of severity and higher priority in receiving treatment. The Status attribute represents the current status of the patient, which can be "Waiting" (0), "Treating" (1), or "Treated" (2). The Pain level attribute represents the pain level of the patient, which can range from 0 to 10, with a lower value indicating lower pain.

The Day coming attribute represents the day of the simulation when the patient arrived, and the Time coming attribute represents the time of the day when the patient arrived. The Length stay in ER attribute represents the number of minutes the patient needs to stay in the ER bed, which is initially generated randomly based on the acute level of the patient's condition. This value starts counting down once the patient receives a bed and becomes 0 when the patient is discharged.

The Priority score attribute is a calculated score used to prioritize queuing patients. It is based on the logic of the Acute level, where a smaller score value indicates a higher priority for the patient. The Day assigned and Time assigned attributes represent the day and time when the patient received a bed, respectively. The Day discharged and Time discharged attributes represent the day and time when the patient was discharged from the ER bed, respectively. The Satisfaction score attribute represents the satisfaction score of the patient, which is initially set to 5.0.

The Patient class also includes several methods. The Get Methods allow retrieving various attributes of the patient, such as age, acute level, pain level, status, length stay in ER, priority score, day coming, time coming, time assigned, time discharged, and satisfaction score. These methods return the respective attribute values or calculated values.

The Set Methods allow setting various attributes of the patient, such as priority score, status, length stay in ER, time coming, day assigned, time discharged, day discharged, total wait time, and satisfaction score. These methods set the respective attribute values based on the provided input.

The Patient class also includes a method for calculating the cumulative wait time of the patient at a specific point in time in the simulation environment when the function is called. This method helps to track the total wait time of the patient during the simulation.

Overall, the Patient class inherits attributes and methods from the Person class and provides functionality for representing and manipulating patient data in a simulated emergency room setting. For further information, refer to Class Patient in Appendix Class Definition Table.

3. Class: Emergency Room. The ER class includes two attributes: Patients (list) and Open Bed (int). The Patients attribute is a list that stores patient objects whose status is either "Waiting" or "Treating". This list does not include patients who have left the ER after being discharged or leaving without being seen (LWBS). The Open Bed attribute is an integer that represents the total number of open beds at a specific point in time, indicating the availability of beds for incoming patients.

The ER class also includes several methods. The Get patients method returns a list of all current and past patients who have visited the ER, including patients who have been discharged or left without being seen. The Get patient method takes a patient ID as input and searches for and returns the specific patient object associated with that ID. The Get open beds method returns the current number of available beds in the ER.

The Bed taken method deducts the total number of available beds by 1, indicating that a bed has been assigned to a patient. The Bed discharged method adds 1 to the total number of available beds, indicating that a patient has been discharged and the bed is now available again.

The Count patients method returns the total number of patients in the ER, including patients who are currently waiting, being treated, or have been discharged. The Count waiting method returns the total number of patients who are currently in the queue, waiting for a bed or treatment. The Count treating method returns the total number of patients who are currently receiving treatment in the ER. The Count treated method returns the total number of patients who have been discharged from the ER after receiving treatment.

These methods provide functionality for managing and tracking the status and availability of beds and patients in the simulated ER environment, allowing for efficient patient flow and resource management.

4. Class: Queue. The Queue class is a data structure used to store patient objects in a list whose status is "Waiting". The Queue attribute is a list that represents the queue of patients waiting to be processed.

The Queue class also includes three methods: Enqueue, Dequeue, and Push to End. The Enqueue method adds a patient to the end of the queue, allowing new patients to join the queue and wait for processing. The Dequeue method removes the patient from the front of the queue, indicating that the patient has been processed and is no longer waiting. Push to End method utilize both Enqueue and Dequeue method to push a patient in the front of the queue to the end of the queue.

The Queue class and its methods are commonly used in computer programming and simulation modeling to manage and track the order in which items are processed or serviced. The Enqueue, Dequeue, and Push to End methods provide functionality for adding and removing items from the queue, allowing for dynamic updates as new items arrive or existing items are processed. In the context of healthcare management, managing a queue of items in a sequential manner is critical to the overall efficiency and fairness of system performance.

4.2 Operations at Patient-level

4.2.1 Patient Operation

This module is essential for simulating the arrival and triage process of patients in the ER setting.

(i) New Patient. The New Patient method creates a patient object by randomly generating values for various attributes within predetermined ranges. These attributes represent the initial conditions of a visiting patient after triage. The attributes generated in this method include Age (ranging from 1 to 100), Acute level (ranging from 1 to 5), Pain level (ranging from 0 to 10), Code (with a default value of "P" for Patient), Day coming, Time coming, and Priority Score.

The Age attribute represents the age of the patient, the Acute level indicates the severity of the patient's condition, and the Pain level represents the reported pain level of the patient. The Code attribute is a code assigned to the patient, with the default value set to "P" for Patient. The Day coming and Time coming attributes represent the day and time of the patient's arrival at the ER, respectively. The Priority Score attribute is calculated based on the values of other attributes and represents the priority level of the patient for further medical attention.

(ii) Priority Score. Priority Score function plays the key role in the simulation. Determining how a waiting patient is prioritized directly affects the outcomes of the simulation. The score, in effect, is calculated using the initial conditions of each incoming patient and provides a metric for bed assignment. In the current version, a Priority Score is a function of Acute Level and Vulnerable Group, where Acute Level is the dominant determinant of the score value. Adding/deleting/editing attributes, or modifying the weight per attribute in the equation are approaches user can take to adjust the priority score of patients in queue. Below is the formula in use:

$$\text{Priority Score} = \text{Acute Level} + \text{Vulnerability Score} \tag{1}$$

In our simulation, the age of all patients is divided into three groups. The division of age group is in accordance with the notion of vulnerability of the patients based on their age: "High Risk", "Medium Risk", or "Low Risk". The high-risk group contains patients whose age is less than or equal to 10 or whose age is larger than or equal to 65. The medium-risk group includes patients whose age is between 11 to 21. Patients with the age from 22 to 64 are generally categorized as "Low Risk". The high-risk group will be assigned a score of 0.1, followed by the medium and low-risk groups with scores of 0.2 and 0.3, respectively. Herein is the logic behind the Priority Score provided by our formula: the higher the score, the lower the priority.

(iii) Time in ER. This is estimated based on the acute level. The user determine the possible range of minutes a patient would stay in the ER bed once moved from waiting to treatment

4.2.2 Satisfaction

Modern healthcare reform efforts have been emphasizing on patient-centered care where patient satisfaction plays the key role in reflecting the performance of healthcare services. According to Boudreaux and O'Hea (2004), "satisfied patients may be more compliant with their medical regimens, suggesting that satisfaction may be an important component in promoting health and well-being." [24] Boulding etl al.'s study later endorsed that in hospitals for acute myocardial infarction and pneumonia, higher overall patient satisfaction is strongly associated with "30-day risk-standardized hospital readmission rates after adjusting for clinical quality" [25]. Simultaneously, low mortality index was discovered to be consistently correlated with the higher satisfaction across 9 of 10 HCAHPS domains [26]. Satisfied patients, furthermore, "directly impact the financial viability of an institution" and promote the integrity of local medical systems since it alleviate the situations of patients "choosing to shop for healthcare services elsewhere" [24]. In the free competitive healthcare system such as that of United States, patient satisfaction is "The Indispensable Outcome" for quality improvement in the healthcare management context [27].

A patient's satisfaction of care is a result of individual experience in different services and departments of a hospital. As a "front door to services", a patient's experience at the emergency department paves the foundational impression on a hospital operation and its staffs. Unfortunately, most ED visits "frequently occur during times of stress and uncertainty for the patient and in an ED care environment that faces a myriad of challenges" that consequently resulted in low satisfaction and medical outcomes [28]. Understanding factors affect patient satisfaction during ED visits and studying approaches to boosting the patient experience at ED, therefore, have been main areas of focus of emergency care practitioners and leaders.

This module subsequently focuses on calculating the satisfaction score of a patient's ER visit solely base on (1) their pain level and (2) their wait time before receiving treatment. The highest possible score is 5.0, and the lowest possible score is 0. The calculation of satisfaction score is only called one time in the program when (1) a patient get assign to a bed or when (2) the wait time of a patient has reached the assigned maximum wait time [2] (or LWBS). Since the default score is 5.0, a patient with a 5.0 satisfaction score can be one of the following: (a) an individual who has not yet received a bed, (b) a patient who received a bed with a total wait time equals to 0, or (c) a patient who received a bed with a total wait time less than the assigned benchmark D. [3].

Overall, there are two mainstream approaches for point calculation: reward and deduction. In our study, the core logic of determining points makes it more intuitive to facilitate the deduction method. Such logic is the more a patient waits, the lower the satisfaction score they would report. The deducting factor is based on a patient's pain level. For instance, patients A and B both have the same number of waited minutes. However, since patient A reported a higher pain level compared to that of patient B during triage, the final satisfaction score surveyed patient A will be lower than the score of the other patient.

The deducting factor, or dpoint, is a discrete value that corresponds to a different level of pain used in deducting a patient's satisfaction score. We refer the reader to the flowchart of Figure 2 for an overview of how dpoint is determined.

To calculate the Satisfaction Score, we propose two methods. In the first method, the deduction of satisfaction starts after arrival. The formula in use for this method is:

$$\text{Satisfaction Score} = 5 - (\text{Total Wait Time}/5) \times \text{dpoint} \tag{2}$$

The second method gives a grace period before starting the deduction. The user determines Benchmark D at the initial condition before running the simulation where Benchmark D is the number of waiting minutes after which the deduction of the Satisfaction Score starts. The formula for this method is as followed:

$$\text{Satisfaction Score} = 5 - (\text{Total Wait Time - Benchmark D}/5) \times \text{dpoint} \tag{3}$$

The minuend 5 in either formula refers to the highest possible satisfaction score, which is also the default score of an incoming patient. The divisor 5 of either total wait time in full or after a grace period indicates

[2] Refer to section 4.5 - Initial Conditions for definition of maximum wait time

[3] Refer to section 4.5 - Initial Conditions for the definition of Benchmark D

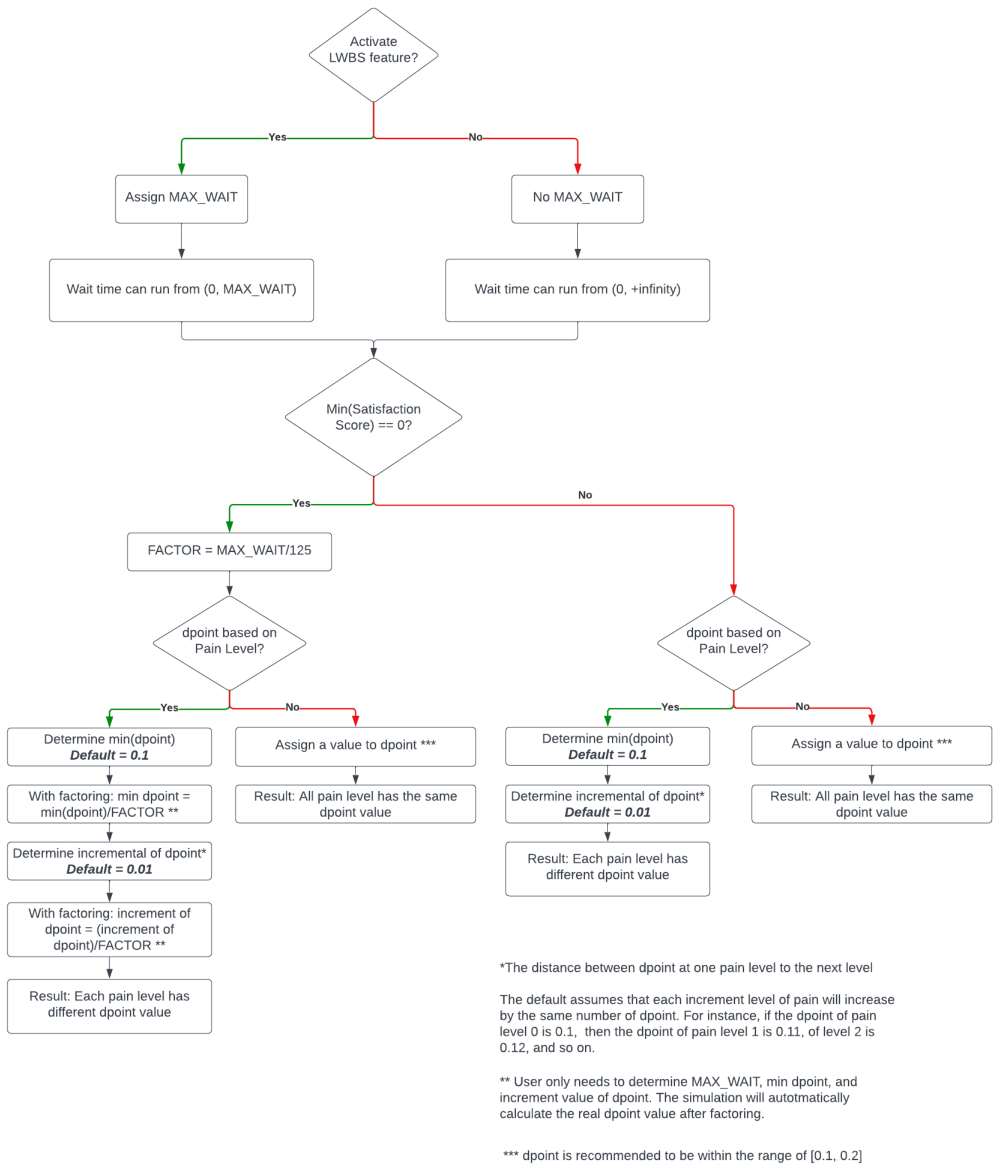

Figure 2: Calculation of dpoint

that the total satisfaction score is deducted per five-minute period. Unless the user wants to change the maximum satisfaction score, the minuend 5 should be held constant. The divisor 5, however, can be adjusted in accordance with the user's need. Noted that if this value is altered, the user also needs to adjust how FACTOR is calculated in determining the point if one seeks to keep the floor value of Satisfaction Score as 0.

It is, in fact, up to the discretion of the user to determine the floor and ceiling value of the satisfaction score. In our study, setting the lowest possible score equal to zero facilitates our analysis when the LWBS feature is activated. Since we have assigned the maximum wait time a patient could wait before leaving, there always exists a floor where the satisfaction score cannot go lower. Therefore, providing the range for our satisfaction score eliminates the need to determine the descriptive statistics for our score and simplifies the score interpretation. For instance, using the range 0-5, we can categorize patients' scores as "Very Unsatisfied", "Unsatisfied", "Neutral", "Satisfied", and "Very satisfied".

4.3 Operations at ER-level

(i) ER Operation. This module includes functions that control the operations in the emergency room. It includes three major functions: "Release Patient", "LWBS", and "Assign Bed".

The first function, "Release Patient", is called once per minute in the simulation environment. It begins by deducting one minute from the total length of stay for all patients currently receiving treatment in the ER. Subsequently, the function identifies patients whose total length of stay has reached or fallen below 0, indicating that they have been treated, and proceeds to discharge them. Upon discharging a patient, the function performs the following tasks: (1) sets the time of discharge for the discharged patient, (2) sets the day of discharge for the discharged patient, (3) changes the status of the patient from "Treating" to "Treated", (4) updates the count of discharged patients for the hour, and (5) updates the number of available beds in the ER.

The second function, "LWBS" (Left Without Being Seen), is also called once per minute in the simulation environment. It searches for patients who have been waiting for a time period exceeding the assigned maximum wait time. Subsequently, it updates the status of these patients to "Left Without Being Seen" and sets their satisfaction score to 0.

The third function, "Assign Bed", is called whenever a bed is available and the queue has at least one patient waiting. It assigns a selected patient to an open bed by setting their status from "Waiting" to "Treating", and performs the following additional tasks: (1) sets the patient's "Time in ER" , (2) records the patient's day and time of receiving a bed, (3) calculates the satisfaction score of the patient, and (4) removes the selected patient from the queue.

(ii) Queue. This module includes functions where the queue is created and how the waiting patients are selected from the queue. The module includes two major functions: "Create Queue" and "Select From Queue".

"Create Queue" establishes a list, referred to as the "Queue", which serves as a repository for patient objects whose status is currently designated as "Waiting". This function is invoked once per minute within the simulation environment.

"Select From Queue", on the other hand, constitutes the primary algorithm that governs the decision-making process for selecting waiting patients from the Queue to be assigned to available beds. It determines the criteria and rules used for prioritizing patients and allocating beds based on various parameters such as severity of condition, waiting time, or other relevant factors.

4.4 Selection Algorithms

"Select From Queue" function is the heart of the program. In this paper, we introduce eight algorithms for bed assignments.

1. Naming Convention

P and Q codes are only present after a dash ("-") in the method name, providing a supplemental description of the approach used by the algorithm for selecting patients. Such additional description is necessary only if the paired comparison of all sub-methods results in at least one pair with different approaches.

Code	**Type**	**Description**
FIFO	Method category	First in first out
C	Method category	Combination of P and Q approaches
A	Method category	Alternate approaches in the same queue after Benchmark W
Z	Method category	Rotation between probes
Integer	Sub-method ordering	Order of the method within the same category (A,C,Z)
P	Approach	Supplementary description denotes that the approach used for selection is based on the logic of Priority Score
Q	Approach	Supplementary description denotes that the approach used for selection is based on the logic of FIFO and/or LIFO

Table 1: Naming Convention of Algorithm

	A1	**A2**
A1	T	F
A2	F	T

Explain: PP ≠ PQ

	C1	**C2**
C1	T	T
C2	T	T

Explain: PQ = PQ

	Z1	**Z2**	**Z3**
Z1	T	T	F
Z2	T	T	F
Z3	F	F	T

Explain: PQ ≠ Q

Figure 3: Paired Comparison of the Supplementary Codes between Sub-methods

For example, in the method category A, A1 and A2 have different supplementary codes, "PP" and "PQ", respectively. Initially, both A1 and A2 use Priority Score as the rule for patient selection, which is indicated by the first "P" in the supplementary codes. However, once a patient's wait time reaches the benchmark W, A2 switches to prioritizing patients using the logic of FIFO and LIFO, which explains the letter "Q" in the supplementary code. On the other hand, A1 continues to use Priority Score as the rule for bed assignment, resulting in the second "P" in the supplementary code.

In contrast, the paired comparison of all sub-methods in the C category shows no difference in the approaches used. Both C1 and C2 algorithms, if described using supplementary codes, would be denoted as "PQ". This indicates that both C1 and C2 use Priority Score as the criteria until the benchmark W is reached, and then apply FIFO (C1) or LIFO (C2) after benchmark W. As a result, there is no need to include supplementary codes in the names of C1 and C2.

The number of letters in the supplementary description subsequently signals whether the algorithm applied the logic of Benchmark W. When there are two letters in the supplemental portion of the method name, we understand that such method has Benchmark W as part of its mechanism. The first letter code of the supplemental description corresponds to the approach applies before the rule of Benchmark W kicks in, followed by the second letter code which indicates the approach in effect once the requirement of Benchmark W is met. See Table **??** for the description of the codes.

2. First-in-first-out (FIFO)

   FIFO skips the presence of Priority Score and simply prioritizes patients based on who comes first. For such reason, patient IDs play the key role since each ID shows the timing order of a visiting patient. Each round of bed assignment will select the minimum value of ID in the queue, then find the patient with matching ID to start treatment.

3. C1: Priority Score and FIFO

   The algorithm of C1 prioritizes patients for bed assignment based on multiple criteria. Firstly, patients with the longest wait time, whose wait time exceeds the Benchmark W, are given the highest priority. Secondly, if no patients meet the above condition, the priority score of each patient is considered, where patients with lower priority scores are given higher priority for bed assignment. Lastly, in case multiple patients have the same lowest priority score, the patient with the smaller id is given higher priority. In other words, C1 algorithm combines the use of priority score and First-In, First-Out (FIFO) principle as the criteria for bed assignment.

4. C2: LIFO, Priority Score, and FIFO

   The algorithm of C2 is similar to C1 with a little twist in how it selects patient once Benchmark W criteria is met. Firstly, based on the value of Benchmark W, the algorithm selects the patient with the highest ID number among all the waiting patients in queue. If no such patient is found, the algorithm prioritizes patients based on their priority score, where patients with lower priority scores are given higher priority. Additionally, in case multiple patients have the same lowest priority score, the patient with the smaller ID is given higher priority for bed assignment.

   In other words, before any queuing patient's wait time hits Benchmark W, C2 utilizes Priority and FIFO as the criteria for bed assignment. Once the patient with the longest wait time has their wait time equal to or larger than the assigned benchmark for waiting, the algorithm switches to Last-come-first-serve (LIFO) as a way to prioritize patients.

   Noted that both C1 and C2 use Benchmark W as the foundation for constructing its first rule. This is important since if we only use Priority Score as the criteria, then in scenario when the number of waiting patients is significantly larger than the number of open beds, the patients whose score always higher than the rest of the group will never get selected[4]. Benchmark W helps avoid such bias by limiting to which extent Priority Score should be applied in the selection process.

   Another important note is that the rule of Benchmark W only presents when there exists at least one patient whose wait time reached the benchmark. If the queue at the time meets the criteria, then after some rounds of bed assignments no patients whose wait time in queue met the rule of Benchmark W, the algorithm will get back to the regular rules as stated (in the case of C1 and C2 the selection process will be continued using priority score and order in line). This mechanism stays true for all of the methods using Benchmark W we present in this paper.

5. A1-PP: Alternate Priority

   The algorithm of A1 for patient selection in bed assignment is defined as follows. First, the algo-

[4]Recall that the higher the score the less priority a patient will be considered in the selection process

rithm checks if there is any patient whose wait time exceeds the Benchmark W. If such a patient is found, the algorithm alternates between selecting a patient with the highest priority score and a patient with the lowest priority score at each instance of bed assignment. If no patient exceeds the Benchmark W, the algorithm prioritizes patients based on their priority score, where patients with lower priority scores are given higher priority. In case multiple patients have the same priority score, the patient with the smaller ID is given higher priority for bed assignment.

For example, if none of the waiting patients has their wait time reached Benchmark W, we will assign bed based on the priority score and the order in queue, or ID, as needed. If the patient who has been waiting the longest has their wait time equal to or exceed Benchmark W, we proceed to change our way of patient selection. For the current round of bed assignment, we will pick the one with the smallest value of priority score. In the next round, we switch to choose the individual whose score is the highest. This rotation between selecting the minimum and maximum score continues till no patients in queue has their wait time equal to or exceed Benchmark W. Again, when the criteria of Benchmark W applies and there are multiple patients whose score are the same, we proceed with the person with smaller ID value.

6. A2-PQ: Alternate FIFO/LIFO

   The algorithm of A2 for patient selection in bed assignment is defined as follows. First, the algorithm checks if there is any patient whose wait time exceeds the Benchmark W. If such a patient is found, the algorithm alternates between selecting a patient with the longest wait time and a patient with the shortest wait time at each instance of bed assignment. If no patient exceeds the Benchmark W, the algorithm prioritizes patients based on their priority score, where patients with lower priority scores are given higher priority. In case multiple patients have the same priority score, the patient with the smaller ID is given higher priority for bed assignment.

7. Z1-PQ: Alternate with Two Probes

   The algorithm of Z1 for patient selection in bed assignment is defined as follows. If there exists a patient whose wait time exceeds the Benchmark W, the algorithm divides the patients in the queue into two probes. It then alternates between selecting a patient with the longest wait time in the first probe and the second probe. If the total number of patients at the point of division is odd, the second probe will have one more patient compared to the first probe. If no patient exceeds the Benchmark W, the algorithm prioritizes patients based on their priority score, where patients with lower priority scores are given higher priority. In case multiple patients have the same priority score, the patient with the smaller ID is given higher priority for bed assignment.

   For Z1, user does not need to specify the number of probes in Config module. For Z2 and Z3, however, user is required to indicate the number of probes (n) before running the program. User can also specify the order of probes, though it is optional.

   By default, n is set at 4. The probes are ordered by interleaving the first and last, then the second and second-to-last, and so on, in an array of continuous numbers ranging from 1 to n. For instance, if the number of probes is set as 7, the order of probes will be [1,7,2,6,3,5,4]. For our default setting, the order of probes is [1,4,2,3]. The purpose of using such method in calculating the order of probes is discussed in section 5 - Discussion.

   When the number of patients in queue is less than n, one patient will be placed in each probe up till the total number of patients. For example, if the number of probes is 5, and there are 3 patients, then probe 1 to 3 will have one patient and the last two probes will be empty.

   When the number of patients in queue is larger than n and the number of patients in queue cannot be fully divided by n, the first number of probes corresponding to the remainder of the division will each have one more patient compared to the last probe. For example, at the current round of bed assignment, we have 43 patients and $n = 5$. The remainder of dividing 43 by 5 is 3. As a result, the first three probes will have 9 patients, and the last two probes will have 8.

8. Z2-PQ: Alternate with Multiple Probes

   The algorithm of Z2 for patient selection in bed assignment is defined as follows. If there exists a patient whose wait time exceeds the Benchmark W, the algorithm divides the patients in the queue into n probes. It then takes turns in selecting a patient with the longest wait time in each probe using the provided probe order. If no patient exceeds the Benchmark W, the algorithm prioritizes patients based on their priority score, where patients with lower priority scores are given higher priority. In case multiple patients have the same priority score, the patient with the smaller ID is given higher priority for bed assignment.

9. Z3-Q: Alternate FIFO with Multiple Probes

   In every round of bed assignment, Z3 will divide the patients in queue into n probes. Then, the algorithm takes turn in selecting an individual with the longest wait time in each probe using the provided probe order. With that being said, Benchmark W is not part of Z3 logic.

4.5 Initital Conditions

All conditions are modifiable, including but not limited to the listed below.

Module	Name	Description
Config	Max Days	The number of days the simulation runs
Config	Max Patients	Maximum number of new patients visit the emergency room in a day
Config	Max Beds	Total number of beds
Config	Benchmark W	To be used in Queue Selection - C1 and C2 method (Queue module). When a queuing patient whose wait time reaches this benchmark of wait time (in minutes), they will be prioritized in the process of bed assignment
Config	Benchmark D	To be used in calculating Satisfaction Score - 'Deduction starts after Benchmark D' method (Satisfaction module). This is the grace wait time period (in minutes) after which the satisfaction of queuing patients starts to be deducted
Config	Max Wait	The maximum minutes a patient will wait before left without being seen (LWBS)
Config	Probes	To be used in method Z2 and Z3 of Select Queue algorithm
Satisfaction	Method of calculating Satisfaction[4]	
Queue	Method of Queue Selection[4]	
PatientOps	Time stay ER[4]	
PatientOps	Priority score calculation[4]	

4.5.1 Output

(i) Printing Output. The printing module displays output at the terminal and stores output into files for exporting

(ii) Exporting Output. The export function can be found under the Auxiliary module. Datasets are exported as csv files using white space as the delimiter. Exported csv files can be found in the Output folder within the same working directory. The default path indicated in Auxiliary module for exporting files requires that "EmergencyRoom", "PatientList", and "Satisfaction" folders must be present within the Output folder. An error will raise at the end of one simulation if the user runs the program without editing path or adding the folders beforehand.

(iii) Reading Output. Refer to Appendix B for the output description.

4.6 Experiment Setups

We set the initial condition for the simulation to run for 15 days, as this provides ample time to observe the behavior of the algorithm in different patient throughput situations. To create three scenarios of overcrowding

[4]Refer to section 3.1 for detail

levels (High, Medium, and Low), we configure the number of beds in the ER department. All other conditions remain the same across the scenarios. The High level of overcrowding is set at 10 beds, followed by 12 beds for Medium, and 15 beds for Low. The tables presented below provide a detailed overview of the configuration of initial conditions.

Name	**Value**
Max day	15
Max patient	5
Max bed	10 (High), 12 (Medium), 15 (Low)
Max wait	720
Benchmark W	240
Benchmark D	30
Probes	4
Probe Order	Default
Method of Calculating Satisfaction	2

Table 3: Set up in Config Module

Age Group	**Priority Score**
Less than 11 or more than 66	+0.1
Between 11 and 21	+0.2
Other	+0.3

Table 4: Set up in PatientOps Module

Acute Level	**Time in ER**
1	[1, 1200]
2	[1, 900]
3	[1, 600]
4	[1, 240]
5	[1, 120]

Table 5: Set up in PatientOps Module

4.7 Evaluation Metrics

Vanbrabant et al. (2019) provides a comprehensive review of KPIs used to evaluate the operation of ED simulations [18]. Based on compiled data presented in Vanbrabant's finding, we design a quantitative framework which fits best to our study purpose. As we seek to observe the efficiency in the process and satisfaction of visiting patients, the amount of wait time, the number of LWBS patients, and average satisfaction are the key indicators of our metrics. Subsequently, our core evaluation encompasses answering the following three questions:

1. Which method yields better results in reducing patient wait time?
2. Which method yields better results in reducing the number of LWBS patients?
3. Which method yields better results in boosting the satisfaction of patients?

5 Results

5.1 Wait Time

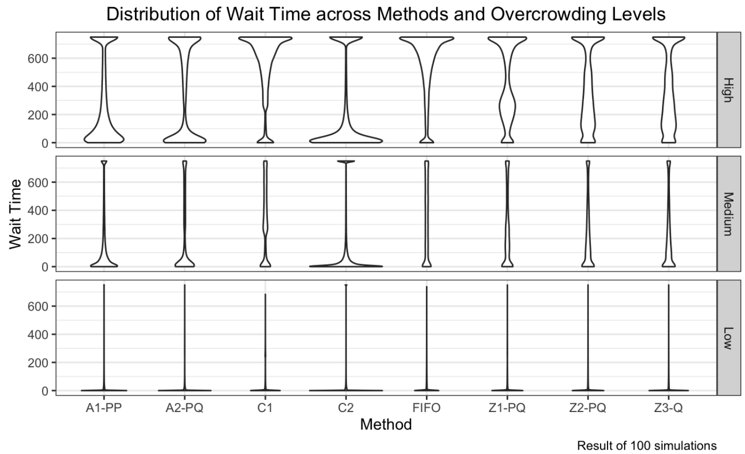

Figure 4: Distribution of Wait Time across Methods and Overcrowding Levels

In Figure 4, we present the violin plots indicating distribution of wait time across methods and overcrowding level. The wider section of a violin indicates the higher probability of wait time that patients experienced. One notable observation is that methods with similar approaches tend to exhibit similar distributions. For example, the distribution of C1 and FIFO methods appears to be relatively identical across different levels of overcrowding. A similar pattern can also be observed in the distributions of Z2 and Z3 methods.

When the level of overcrowding is high, C2 shows the best performance among the methods, with the majority of patients experiencing less than 400 minutes of wait time. A1, A2, Z2, and Z3 follow C2 in terms of wait time performance. However, it is worth noting that C2 also has the highest number of patients whose wait time reaches the maximum of 720 minutes when the overcrowding level is medium and low. Apart from C2, the lower the level of overcrowding, the less noticeable the difference in distribution between methods.

5.2 LWBS

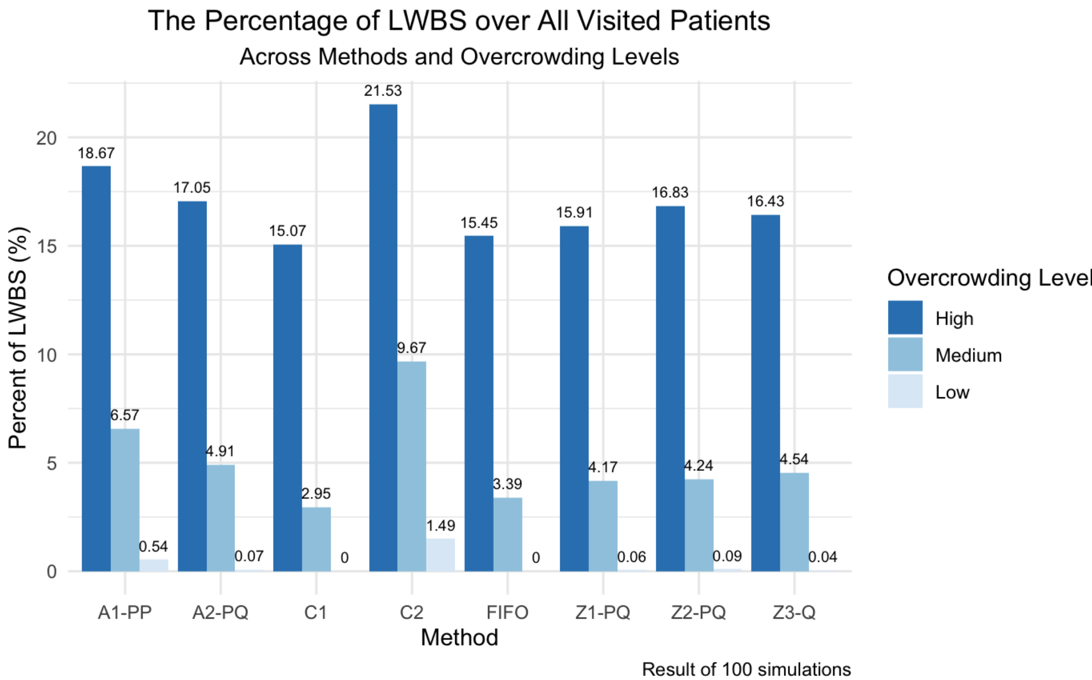

Figure 5: The Percentage of LWBS across Methods and Overcrowding Levels

Figure 5 shows the percentage of LWBS across methods and overcrowding. Observation of figure 5 reveals an important pattern between the variation of overcrowding scenarios and the proportion of patients who leave before treatment. To be more specific, the percentage of LWBS patients decreases proportionally with the level of overcrowding, indicating that the ranking of methods with the highest to the lowest percentage of LWBS remains consistent regardless of overcrowding levels. When sorting the results in ascending order, C1 is followed by FIFO, Z1, Z2, Z3, A2, A1, and C2.

This finding highlights an interesting observation: methods that perform well in terms of reducing median wait time often come at a cost. Algorithms such as C2, A1, and A2 prioritize boosting the median wait time of patients in the queue, but as a trade-off, they may end up forgoing more patients whose wait time has been long or exceptionally long compared to others. To gain a deeper understanding of which patient priority groups are more likely to be dismissed by each method, we observe the percentage of LWBS patients across different priority groups. Our current priority score formula places minimal weight on the Age attribute, so priority groups are primarily determined by the Acute Level assigned during triage. The priority group of a patient, in other words, is the value of their truncated priority score.

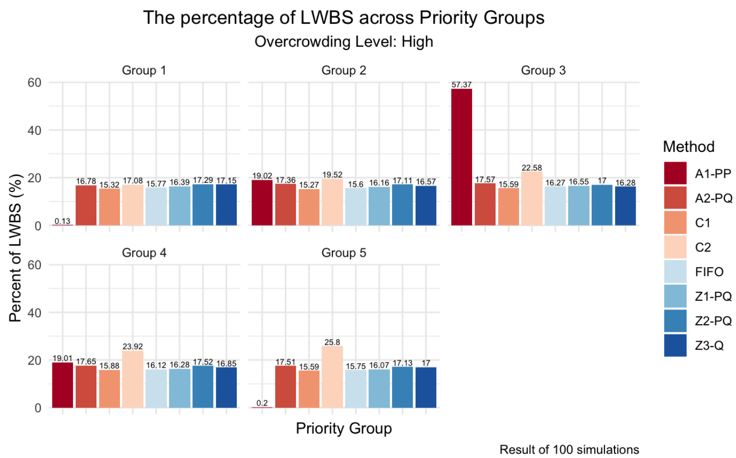

Figure 6: The Percentage of LWBS across Priority Groups with High Level of Overcrowding

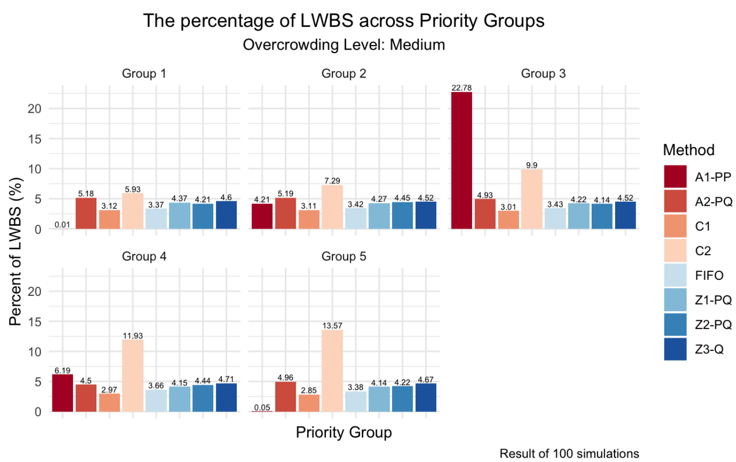

Figure 7: The Percentage of LWBS across Priority Groups with Medium Level of Overcrowding

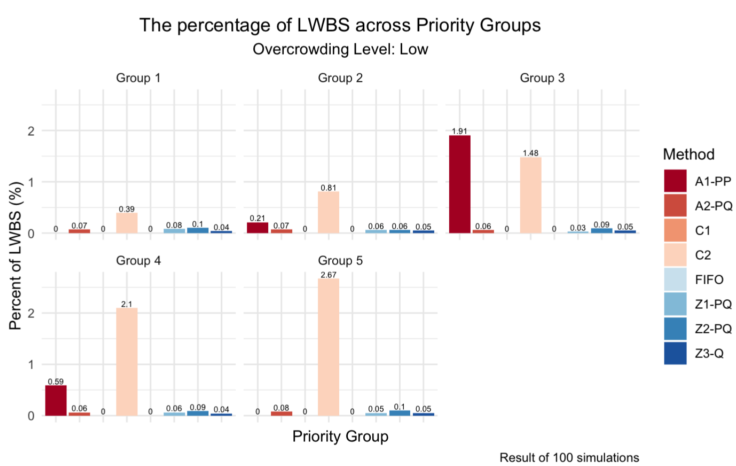

Figure 8: The Percentage of LWBS across Priority Groups with Low Level of Overcrowding

Figure 6, 7, and 8 break down the percentage of LWBS patients resulted by each method into five priority groups when the overcrowding level is high, medium, and low respectively. We group methods that exhibit similar trends to discern variations in the percentage of LWBS patients across priority groups amidst varying levels of overcrowding. Specifically, Group I includes methods A1 and C2, while Group II comprises C1 and FIFO. Lastly, Group III encompasses A2, Z1, Z2, and Z3.

Group I represents the category of "extreme" methods, which exhibits either the best or worst performance across priority groups and overcrowding levels. For instance, A1 consistently demonstrates the lowest percentage of LWBS patients within the first and last priority groups. Similarly, C2 displays the highest or second-highest (after A1) percentage of LWBS patients across all priority groups and overcrowding levels, with the exception of high overcrowding in the first priority group. When the overcrowding level is high, as depicted in figure 6, C2 ranks third in the first priority group for patients left without treatment, at 17.08 percent, preceded by Z2 with a close margin at 17.29%, and Z3 at 17.15 percent.

Group II, also referred to as "the pair," comprises C1 and FIFO, which consistently produce similar outcomes across all priority groups and overcrowding levels. However, their disparities become more apparent only during medium overcrowding scenarios, as observed in figure7, albeit with a minimal margin ranging from 0.7 percent to 0.3 percent difference.

Group III includes the remaining methods, which demonstrate comparable percentages and trends across priority groups and overcrowding levels. These methods consistently demonstrate LWBS patient percentages ranging from 16 to 17 percent during high overcrowding situations (Figure 6), 4 to 6 percent during medium overcrowding situations, and 0.05 to 0.08 percent during low overcrowding situations.

5.3 Satisfaction

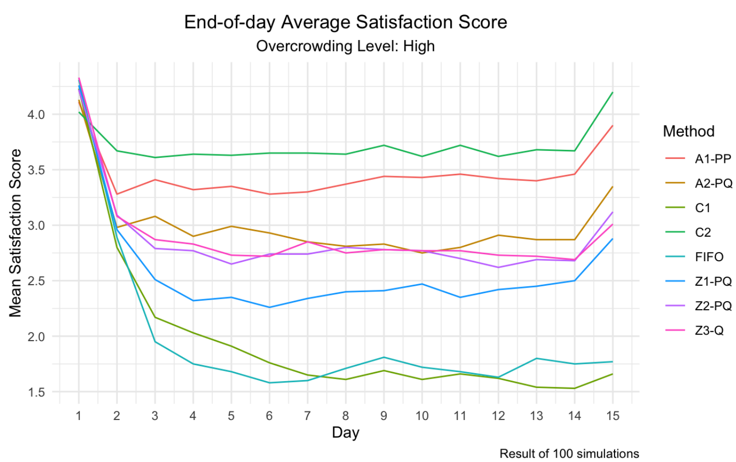

Figure 9: End-of-day Average Satisfaction Score with High Level of Overcrowding

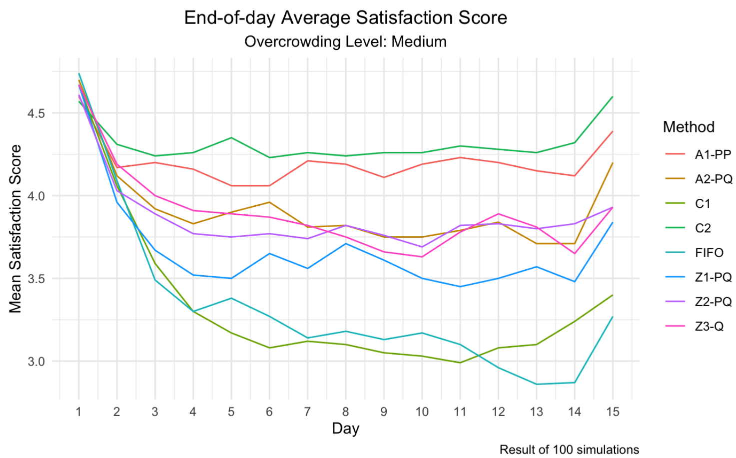

Figure 10: End-of-day Average Satisfaction Score with Medium Level of Overcrowding

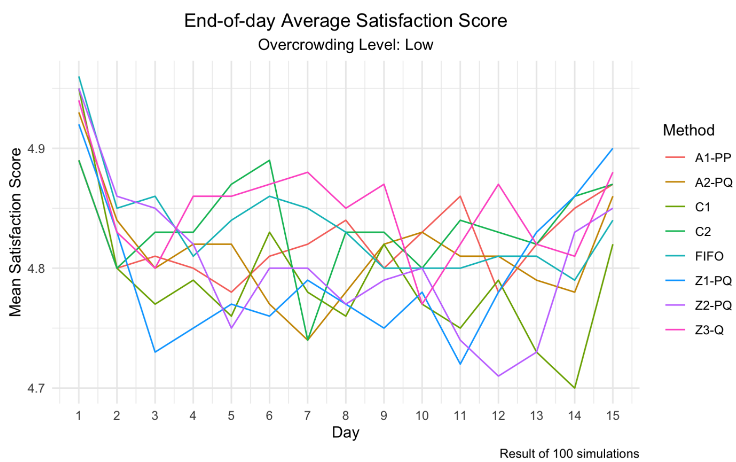

Figure 11: End-of-day Average Satisfaction Score with Low Level of Overcrowding

Figure 9, 10, and 11 depicts the end-of-day average satisfaction score across methods and levels of overcrowding. The categorization of all methods into three groups remains consistent, as evidenced by the end-of-day average satisfaction scores during high and medium overcrowding levels depicted in figure 9 and 10. Methods within the same group exhibit similar results. Group I has the highest score, followed by group II. Methods in group III show the lowest scores compared to those in other groups. Of all methods, C2 yields the highest average satisfaction score across days, followed by A1. A2, Z2, and Z3 share a similar trend, while Z1 consistently appears to be the lowest in group II during both medium and high overcrowding scenarios. In situations where the overcrowding level is low as demonstrated in figure 11, the difference in average satisfaction score across methods is trivial, with a standard deviation of less than 0.05 between methods.

Figure 12: End-of-day Average Satisfaction Score across Priority Groups with High Level of Overcrowding

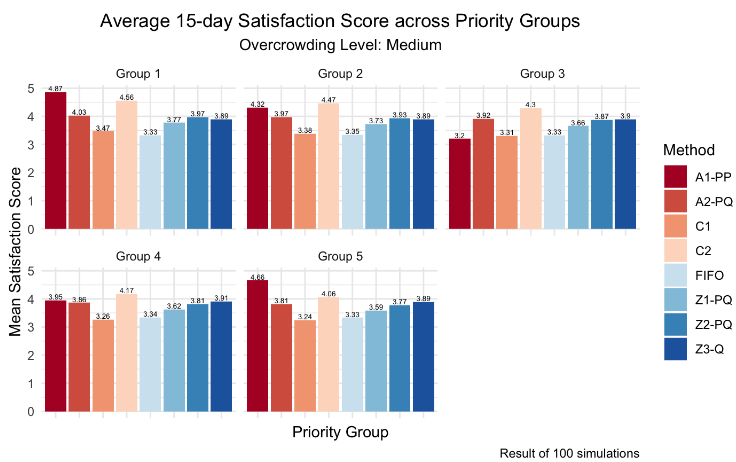

Figure 13: End-of-day Average Satisfaction Score across Priority Groups with Medium Level of Overcrowding

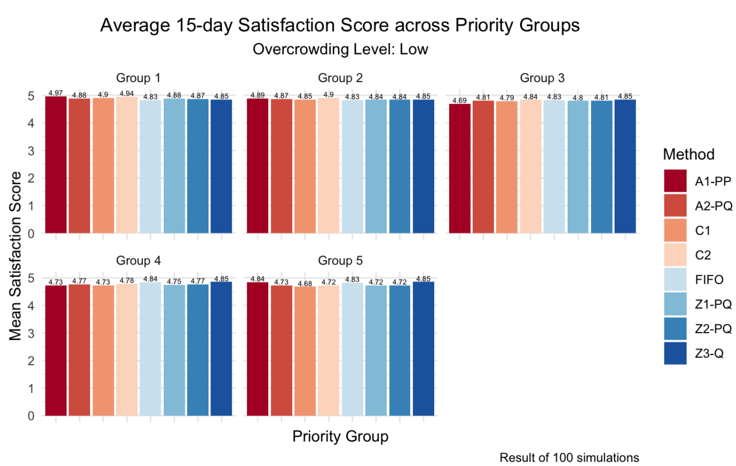

Figure 14: End-of-day Average Satisfaction Score across Priority Groups with Low Level of Overcrowding

Figure 12, 13, and 14 breaks down the end-of-day average satisfaction score across methods and overcrowding scenarios into five priority groups. In situations where the overcrowding level is low as shown in figure 14, the average satisfaction scores across priority groups and methods exhibit comparable results. However, as the overcrowding level increases to medium and high in figure 12 and 13, the differences in average satisfaction scores become more apparent across priority groups and methods. Specifically, A1 demonstrates the highest satisfaction scores for patients in the first and last priority groups, but ranks second for the second and fourth, and shows the worst performance among the third priority group compared to all other methods. In contrast, the performance of C2 remains more consistent across all priority groups and overcrowding scenarios. Both A1 and C2, referred to as "the extremes", display either the lowest or highest performance across priority groups and methods.

The pair of C1 and FIFO exhibit similar results, ranking either as the lowest or second-lowest compared to other methods across priority groups. C1 performs slightly better compared to FIFO in the first priority group, with a small difference ranging between 0.06 to 0.14 percent higher. However, for the remaining priority groups, FIFO consistently outperforms C1 by an average of 0.15 percent.

Methods belonging to Group III depict a consistent trend across all priority groups. In general, A2 demonstrates the best performance within Group III, followed by Z3, Z2, and Z1. However, the order differs moderately in the first priority group, as Z2 yields slightly higher results compared to Z3. As a result, among patients in the first priority group, the ranking of methods within Group III is A2, Z2, Z3, and Z1. Furthermore, the differences between Z1 and the other methods within Group III are more noticeable, with an average difference of 0.3 to 0.4 percent. The differences between A2, Z2, and Z3 are less observable, ranging from 0.06 to less than 0.1 percent on average.

6 Discussion

It has been observed that no singular method surpasses others without trade-offs. When a particular method excels in one performance metric, it tends to yield inferior results in other criteria. C2, for instance, exhibits a notably lower median wait time compared to other methods due to forgoing treatment for a higher number of patients who leave without receiving care. In fact, there is a reverse logic that applies to the relationship between wait time, LWBS, and satisfaction scores. Algorithm with a low median wait time often has a high percentage of patients who left before bed assignment and high satisfaction score. Conversely, operation with higher wait times usually has lower percentages of patients who leave before care and lower average satisfaction from all visited patients.

Noted that such a relationship between wait time, LWBS, and satisfaction score is not static. This is evident during high overcrowding environment, Z3 has a lower percentage of LWBS patients and also a higher satisfaction score compared to Z2. Hence, the influence of one metric on another depends on the context and operational structure of each algorithm in specific scenarios.

The efficacy of a given method nonetheless may differ when applied to varying overcrowding levels or different priority groups of patients. The selection of an appropriate method, therefore, depends on several factors including the specific criteria, priority group, and overcrowding condition being addressed.

6.1 The Extremes

A1 favors patients in the first and last priority groups. Its admission rate is exceptionally high among these two groups with a median wait time often less than 200 minutes across all overcrowding scenarios. However, A1 performs poorly the most for patients in the third priority group, which can be attributed to its operational mechanism. The method selects patients with the highest and lowest priority scores alternately once the criteria for benchmark W is met, resulting in a higher likelihood of neglecting individuals whose scores fall in the middle. This issue is particularly pronounced in situations where the emergency room is highly crowded. The average satisfaction score of A1 across priority groups thus reflects the inherent dilemma of the algorithm.

C2, similar to A1, demonstrates favorable performance in terms of median wait time but at a higher cost.

It exhibits the highest percentage of patients who left without receiving treatment, and the proportion of patients who left before being seen increases across priority groups. C2 prioritizes patients with a higher need for immediate care, potentially foregoing those with less severe conditions. Despite having the highest percentage of LWBS patients as depicted in figure 5, C2 surprisingly maintains the highest or second-highest average satisfaction scores across priority groups and overcrowding scenarios. This can be attributed to C2's operational mechanism, where it switches to prioritizing patients who arrived later or whose satisfaction scores have not been significantly impacted by the wait once the condition for benchmark W is met. Patients who have been waiting longer in the queue, under the operation of C2, may continue to wait longer and eventually leave. As a result, although the percentage of LWBS in C2 is noticeably higher compared to other methods, the low satisfaction scores resulting from LWBS patients are offset by the high to perfect scores contributed by the last-in-queue patients who are selected after the benchmark W. Consequently, the distribution of C2 resembles double disks (Figure 4), indicating that patients may experience either a short or exceptionally long waiting time. In other words, C2 favors "newer" patients in the prioritization process.

6.2 The Pair and Fair

In contrast to C2, C1 lags behind in most of the evaluated metrics, except for the percentage of patients who left without receiving treatment. Remarkably, C1 has the lowest percentage of LWBS patients across priority groups and overcrowding scenarios as shown in figure 5, 6, 7, and 8. However, its median wait time and average satisfaction score are also the poorest among the methods evaluated as shown in figure 4, 9, 10, and 11. The strength of C1, however, lies in its fairness. It prioritizes patients who are most urgent before the wait time of the patient who has been in the queue the longest becomes excessively long. Once the wait time of the patient reaches benchmark W, C1 accommodates them and shifts its priority to patients with longer wait times. This explains why both the median wait time and average satisfaction score resulting from the operation of C1 are inferior compared to other methods.

C1, firstly, prioritizes patients with more acute conditions before the benchmark criteria W is met, which results in longer stays in the emergency room and delays in bed availability. Furthermore, when it comes to medium and high overcrowding situations, patients with less severe conditions may have to wait longer or surpass the benchmark W to receive a bed. However, these patients are also more likely to have shorter stays in the ER, which can contribute to increased bed availability. As a result, C1 does not demonstrate a clear bias towards any specific metrics like time or satisfaction. Its focus is divided between prioritizing patients with the most urgent need for care and those who have been waiting the longest.

A method that shares similar results to C1 is FIFO. The performance yields by both C1 and FIFO are especially identical in the highly overcrowded scenarios. This is because, with the effect of benchmark W, C1 operates under the principles of FIFO. However, subtle differences between the two methods become evident when considering satisfaction scores. Except for the first priority group, FIFO outperforms C1 by a slight margin across different priority groups and overcrowding levels. This can be explained by the higher traffic C1 experienced in its waiting queue before FIFO becomes effective upon reaching benchmark W. FIFO, nonetheless, has moderately higher LWBS rate compared to C1 as shown in figure 5. Although both methods excel in less crowded environments with zero patients leaving before bed assignment, C1 stands out as the only method with a maximum wait time for all patients of less than 700 minutes (Figure 4).

6.3 The "Balanced"

Methods in Group III demonstrate a more balanced performance. Among them, A2 exhibits a distribution pattern that suggests patients may experience relatively shorter or longer waiting times, but less drastic compared to C2 (Figure 4). Despite having a moderately higher percentage of patients who leave without receiving treatment compared to other methods in Group III as demonstrated in figure 5, A2 still maintains a significantly lower rate of LWBS compared to A1 and C2. Additionally, its average satisfaction score ranks only second to methods in Group I across overcrowding situations. A2 can be characterized as a less extreme version of A1 and C2 methods, performing at its best in terms of achieving a balance between wait times and patient satisfaction.

A noteworthy comparison can be drawn between C1, Z1, and Z2, as they share similar operational logic with differences in the number of probes utilized. Specifically, C1 is algorithm when it only has one probe, while Z1 and Z2 employ two and four probes, respectively. It can be observed that as the number of probes increases, the distribution of wait times tends to become more spread out (Figure 4). However, this also leads to a higher percentage of patients leaving before being seen (Figure 5), which in turn increases the average satisfaction score due to shorter wait times for those who are seen (Figure 9, 10, 11). Notably, in the context of an overcrowded ER environment, Z2 exhibits a distribution of wait times that indicates patients under this operation are less likely to wait for more than 600 minutes compared to Z1. However, Z1 demonstrates a lower percentage of patients leaving without receiving treatment at a more consistent basis.

Z3, at the same time, is the mutation of Z2. When the criteria of benchmark W in Z2 is more likely to be activated in highly overcrowded ER scenarios, the result of Z2 is very similar to that of Z3. In highly overcrowding situations, however, Z3 performs better with a lower percentage of LWBS and a higher average satisfaction score across priority groups as demonstrated in figure 6 and 9. When the ER environment is less crowded, Z2 results in a lower percent of LWBS and better satisfaction scores in the first two priority groups of patients (Figure 13). This can be explained as Z2 algorithm selecting patients based on priority score before the rule of benchmark W goes into effect. When the level of overcrowding is lowest, as shown in figure 14, Z3 nonetheless performs better than Z2 across criteria.

6.4 Algorithm Selection

As a result, the selection of the algorithm is based on various factors, including the size of the ER department, the current situation in the ER, statistical information obtained by administrators about visiting patients, and cost-benefit analysis focused on metrics and demographics. In the search for a solution, the design manager must carefully weigh the trade-offs between efficiency, satisfaction, and fairness. As observed, the gain in one area often results in a loss in another.

Another criterion we need to consider is the practicality of a method. Eight different operations are provided, yet not all can be implemented. Among those, C2 stands out as a method that may only be feasible in theoretical studies. Bypassing all patients who have been waiting for the longest and selecting only those who come in last without causing chaos may not be practical. However, the results of C2 provide an important piece of information: LIFO boosts the average satisfaction score, as well as the median wait time for all patients in the queue. In practice, selecting an individual who comes in last is more challenging since this would create tension with the rest of the waiting patients. However, it is feasible to select the first individual of the group who comes in last if all waiting patients are divided into isolated zones. Hence, the logic of Z1, Z2, and Z3 and the calculation of how probes are ordered by default using interleaving operation.

6.5 Limitations

The current version of the simulation does not take into account the interaction between patients with other ER resources whose availability could otherwise affect the results. Medical specialists, scanning devices, and operation rooms, for instance, are resources whose operational speed and vacancy could have impacted the wait time and satisfaction of individual patients.

7 Conclusion

Simulation of the emergency room provides a risk-free environment to quantify the efficiency and satisfaction of different queuing algorithm. However, it is evident that no single approach could perform well in one metric without forgoing another. Choosing the optimal approach for a given situation depends on various factors, such as the size of the emergency department, current conditions in the ER, statistical data on visiting patients, and a cost-benefit analysis that considers relevant metrics and demographics. Results from simulations further enable analytics of ED operations and enhance strategic decisions made by administration of the systems.

Future Directions. Incorporating clinical data could enhance the design of the simulation and make results more system specific. Additional features and classes could be included to replicate the complexity of emergency department (ED) operations, such as healthcare professionals and other infrastructures. At the most basic level, configuring the initial conditions of the ED could provide further insights and outcomes, including but not limited to adjusting values, operational logic, and adding or deleting features, among others.

References

[1] N. C. for Health Statistics, "Percentage of hospital emergency department visit in the past 12 months for adults aged 18 and over," united States, 2019 Q1, Jan-Mar—2022 Q3, Jul-Sep. National Health Interview Survey. Generated interactively: Apr 14 2023 from. [Online]. Available: https://wwwn.cdc.gov/NHISDataQueryTool/ER_Quarterly/index_quarterly.html

[2] C. Cairns, J. J. Ashman, and K. Kang, "Emergency department visit rates by selected characteristics : United states, 2018," Hyattsville, MD, Sep 03, journal Issue. [Online]. Available: https://stacks.cdc.gov/view/cdc/102278

[3] N. Risko, A. Chandra, T. W. Burkholder, L. A. Wallis, T. Reynolds, E. J. Calvello Hynes, and J. Razzak, "Advancing research on the economic value of emergency care," *BMJ Glob. Health*, vol. 4, no. Suppl 6, p. e001768, Jul. 2019.

[4] A. C. of Emergency Physicians *et al.*, "Emergency department crowding: high-impact solutions," *ACEP task force report on boarding*, 2008.

[5] A. T. Janke, E. R. Melnick, and A. K. Venkatesh, "Hospital Occupancy and Emergency Department Boarding During the COVID-19 Pandemic," *JAMA Network Open*, vol. 5, no. 9, pp. e2 233 964–e2 233 964, 09 2022. [Online]. Available: https://doi.org/10.1001/jamanetworkopen.2022.33964

[6] Y. Ng and S. Lewena, "Leaving the paediatric emergency department without being seen: Understanding the patient and the risks," *Journal of paediatrics and child health.*, vol. 48, no. 1, 2012.

[7] V. Weston, S. Jain, M. Gottlieb, A. Aldeen, S. Gravenor, M. Schmidt, and S. Malik, "Effectiveness of resident physicians as triage liaison providers in an academic emergency department," *West. J. Emerg. Med.*, vol. 18, no. 4, pp. 577–584, Jun. 2017.

[8] L. G. Connelly and A. E. Bair, "Discrete event simulation of emergency department activity: a platform for system-level operations research," *Acad. Emerg. Med.*, vol. 11, no. 11, pp. 1177–1185, Nov. 2004.

[9] J. Kreke, A. J. Schaefer, and M. S. Roberts, "Simulation and critical care modeling," *Current Opinion in Critical Care*, vol. 10, pp. 395–398, 2004.

[10] Wiinamaki and Dronzek, "Using simulation in the architectural concept phase of an emergency department design," in *Proceedings of the 2003 Winter Simulation Conference, 2003.*, vol. 2, 2003, pp. 1912–1916 vol.2.

[11] D. Morgareidge, H. CAI, and J. JIA, "Performance-driven design with the support of digital tools: Applying discrete event simulation and space syntax on the design of the emergency department," *Frontiers of Architectural Research*, vol. 3, no. 3, pp. 250–264, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2095263514000259

[12] D. R. Cooney, M. G. Millin, A. Carter, B. J. Lawner, J. V. Nable, and H. J. Wallus, "Ambulance diversion and emergency department offload delay: resource document for the national association of EMS physicians position statement," *Prehosp. Emerg. Care*, vol. 15, no. 4, pp. 555–561, Oct. 2011.

[13] A. Nahhas, A. Awaldi, and T. Reggelin, "Simulation and the emergency department overcrowding problem," *Procedia Engineering*, vol. 178, pp. 368–376, 2017, relStat-2016: Proceedings of the 16th International Scientific Conference Reliability and Statistics in Transportation and Communication October 19-22, 2016. Transport and Telecommunication Institute, Riga, Latvia. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705817300681

[14] G. Savioli, I. F. Ceresa, N. Gri, G. Bavestrello Piccini, Y. Longhitano, C. Zanza, A. Piccioni, C. Esposito, G. Ricevuti, and M. A. Bressan, "Emergency department overcrowding: Understanding the factors to find corresponding solutions," *J. Pers. Med.*, vol. 12, no. 2, p. 279, Feb. 2022.

[15] R. Carmen and I. V. Nieuwenhuyse, "Improving patient flow in emergency departments with or techniques: A literature overview," *Econometrics: Econometric & Statistical Methods - Special Topics eJournal*, 2014.

[16] D. Sinreich and Y. Marmor, "Emergency department operations: The basis for developing a simulation tool," *Iie Transactions*, vol. 37, pp. 233–245, 02 2005.

[17] P. Bhattacharjee and P. K. Ray, "Patient flow modelling and performance analysis of healthcare delivery processes in hospitals: A review and reflections," *Computers Industrial Engineering*, vol. 78, pp. 299–312, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835214001272

[18] L. Vanbrabant, K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "Simulation of emergency department operations: A comprehensive review of kpis and operational improvements," *Computers Industrial Engineering*, vol. 131, pp. 356–381, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835219301615

[19] M. Defraeye and I. Van Nieuwenhuyse, "Staffing and scheduling under nonstationary demand for service: A literature review," *Omega*, vol. 58, pp. 4–25, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305048315000754

[20] C.-H. Lin, C.-Y. Kao, and C.-Y. Huang, "Managing emergency department overcrowding via ambulance diversion: A discrete event simulation model," *Journal of the Formosan Medical Association*, vol. 114, no. 1, pp. 64–71, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0929664612004627

[21] J. K. Gorski, T. S. Arnold, H. Usiak, and C. D. Showalter, "Crowding is the strongest predictor of left without being seen risk in a pediatric emergency department," *Am. J. Emerg. Med.*, vol. 48, pp. 73–78, Oct. 2021.

[22] U. Hwang, M. L. McCarthy, D. Aronsky, B. Asplin, P. W. Crane, C. K. Craven, S. K. Epstein, C. Fee, D. A. Handel, J. M. Pines, N. K. Rathlev, R. W. Schafermeyer, F. L. Zwemer, Jr, and S. L. Bernstein, "Measures of crowding in the emergency department: a systematic review," *Acad. Emerg. Med.*, vol. 18, no. 5, pp. 527–538, May 2011.

[23] A. T. Janke, E. R. Melnick, and A. K. Venkatesh, "Monthly Rates of Patients Who Left Before Accessing Care in US Emergency Departments, 2017-2021," *JAMA Network Open*, vol. 5, no. 9, pp. e2 233 708–e2 233 708, 09 2022. [Online]. Available: https://doi.org/10.1001/jamanetworkopen.2022.33708

[24] E. D. Boudreaux and E. L. O'Hea, "Patient satisfaction in the emergency department: a review of the literature and implications for practice," *The Journal of Emergency Medicine*, vol. 26, no. 1, pp. 13–26, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736467903002853

[25] W. Boulding, S. W. Glickman, M. P. Manary, K. A. Schulman, and R. Staelin, "Relationship between patient satisfaction with inpatient care and hospital readmission within 30 days," *Am. J. Manag. Care*, vol. 17, no. 1, pp. 41–48, Jan. 2011.

[26] G. D. Kennedy, S. E. Tevis, and K. C. Kent, "Is there a relationship between patient satisfaction and favorable outcomes?" *Ann. Surg.*, vol. 260, no. 4, pp. 592–8; discussion 598–600, Oct. 2014.

[27] M. of the ACEP, "Patient satisfaction," *Emergency Department - Patient Satisfaction Surveys*, Jun 2011. [Online]. Available: https://www.acep.org/imports/clinical-and-practice-management/patient-satisfaction/

[28] J. D. Sonis, E. L. Aaronson, R. Y. Lee, L. L. Philpotts, and B. A. White, "Emergency department patient experience: A systematic review of the literature," *J. Patient Exp.*, vol. 5, no. 2, pp. 101–106, Jun. 2018.

A Source Code

V. Bui. "Emergency Room Simulation," April 2003. [Online]. Available: `https://github.com/vivibui/Emergency-Room-Simulation`

B Class Definition Table

1. Class: Person

Name	Type	Data Type	Description
ID	Attribute	int	Each person coming to the ER will be given an ID based on the order they come in. ID number starts from 1 and is incremented by 1
Code	Attribute	str	The code for patient by default is "P"
Get ID	Method		Return ID
Get code	Method		Return code

2. Class: Patient

Name	Type	Data Type	Description
Age	Attribute	int	Represents the age of the patient, ranging from 1 to 100 years old
Acute level	Attribute	int	Indicates the severity of the patient's condition on a scale of 1-5, with lower values indicating higher severity and priority for receiving treatment. The acute levels are categorized as follows: Level 1 - Resuscitation (immediate life-saving intervention), Level 2 - Emergency, Level 3 - Urgent, Level 4 - Semi-urgent, and Level 5 - Non-urgent
Status	Attribute	int	An integer in the range of [0,2], with the default = 0. A patient will have three statuses: "Waiting" (0) – which is when they are in queue to receive a ER bed, "Treating" (1) – when they receive an ER and receiving treatment, and "Treated" (2) – when they are discharged from the emergency bed
Pain level	Attribute	int	Represents the level of pain reported by the patient, ranging from 0 to 10, with lower values indicating lower pain
Day coming	Attribute	int	An integer denotes the nth day of the simulation that the patient arrived
Time coming	Attribute	time	Represents the time of day (in time datatype) when the patient arrived at the ER
Length stay in ER	Attribute	int	Represents the expected length of stay in minutes for the patient in the ER bed, which is randomly generated based on the patient's acute level at the time of arrival. This value starts counting down once the patient receives a bed and becomes 0 when the patient is discharged

Priority score	Attribute	float	Represents a calculated score used to prioritize the queuing of patients. A lower score indicates a higher priority for the patient, in compliance with the logic of the acute level
Day assigned	Attribute	int	Represents the nth day when the patient was assigned an ER bed
Time assigned	Attribute	time	Represents the time of day (in time datatype) when the patient was assigned an ER bed. If the patient has not yet been assigned a bed by the end of the simulation, "TBD" (To Be Determined) is returned
Day discharged	Attribute	int	Represents the nth day when the patient was discharged from the ER bed
Time discharged	Attribute	time	Represents the time of day (in time datatype) when the patient was discharged from the ER bed. If the patient has not yet received a bed or has received a bed but has not yet completed their length of stay in the ER, "TBD" (To Be Determined) is returned
Satisfaction score	Attribute	float	Represents a calculated score that depicts the level of satisfaction reported by the patient regarding their ER visit. The default value is 5.0
Get age	Method		Return age
Get acute level	Method		Return acute level
Get pain level	Method		Return pain level
Get status	Method		Return status
Get length stay in ER	Method		Return the total minutes have left in the ER bed
Get priority score	Method		Return the priority score
Get day coming	Method		Return the day the patient first comes in
Get time coming str	Method		Return the time the patient first comes in as string
Get time assigned str	Method		Return the time the patient receives a bed as string; return "TBD" if has not yet assigned a bed by the end of the simulation
Get time discharged str	Method		Return the time the patient gets discharged as string; return "TBD" if the patient either has not received a bed or has received a bed but has not yet completed their length of stay in the ER
Get time coming	Method		Return the time the patient comes in (as time datatype)
Get time assigned	Method		Return the time the patient gets assigned a bed (as time datatype)
Get total wait time	Method		Return the total wait time of the patient (in minutes)
Get satisfaction score	Method		Return the satisfaction score of the patient
Set priority score	Method		Set the priority score of the patient
Set status	Method		Set the status of the patient
Set length stay in ER	Method		Set the length of stay in ER of the patient

Deduct time in ER	Method		Deduct the length of stay in ER by one minute each time got called
Set time coming	Method		Set the time the patient comes in
Set day assigned	Method		Set the day the patient comes in
Set time discharged	Method		Set the time the patient is discharged
Set day discharged	Method		Set the day the patient is discharged
Set total wait time	Method		Set the total wait time of the patient
Set satisfaction score	Method		Set the satisfaction score for the patient
Deduct time in ER	Method		Deducts one minute from the length of stay in the ER each time the function is called
Calculation of Wait Time	Method		Calculate the cumulative wait time of the patient at the point of time in the simulation environment the function is called

3. Class: Emergency Room

Name	**Type**	**Data Type**	**Description**
Patients	Attribute	list	A list that stores patient objects whose status is either “Waiting” or “Treating”. This list does not include patients who have left the ER after being discharged or leaving without being seen (LWBS)
Open Bed	Attribute	int	The total number of open beds at a specific point in time, indicating the availability of beds for incoming patients
Get patients	Method		Returns a list of all current and past patients who have visited the ER, including patients who have been discharged or left without being seen
Get patient	Method		Takes a patient ID as input and searches for and returns the specific patient object associated with that ID
Get open beds	Method		Returns the current number of available beds in the ER
Bed taken	Method		Deducts the total number of available beds by 1, indicating that a bed has been assigned to a patient
Bed discharged	Method		Adds 1 to the total number of available beds, indicating that a patient has been discharged and the bed is now available again
Count patients	Method		Returns the total number of patients in the ER, including patients who are currently waiting, being treated, or have been discharged
Count waiting	Method		Returns the total number of patients who are currently in the queue, waiting for a bed or treatment
Count treating	Method		Returns the total number of patients who are currently receiving treatment in the ER

Count treated	Method		Returns the total number of patients who have been discharged from the ER after receiving treatment

4. Class: Queue

Name	**Type**	**Data Type**	**Description**
Queue	Attribute	list	A list stores patient objects whom status is “Waiting”
Enqueue	Method		Add a patient to queue
Dequeue	Method		Remove a patient from queue
Push to End	Method		Bring a patient at the front of the queue to the end of the queue

C Output Description

1. Emergency Room

Name	**Description**
Sim N	The seed number of the simulation
Day	The nth day within the simulation environment
Time	One-hour window of one day within the simulation
New Patients	The number of new patients visiting the ER in the one-hour window
Waiting	The total number of patients in queue
Assigned	The total number of patients who is treating
Discharged	The number of patients who are discharged within the one-hour window
Beds Available End-of-hour	The total number of open beds after bed assignment and discharged at the end of the hour. This value should be equal to the total number of beds that is assigned in the initial conditions minus the value in column ”Assigned”

2. Patient List

Name	**Description**
Sim N	The seed number of the simulation
ID	The ID of the patient
Day C	The nth day within the simulation environment that the patient comes in
Time C	The time of the day within the simulation environment that the patient comes in
Age	The age of the patient
Acute level	The acute level of the patient’s condition

Pain level	The pain level the patient reported
Priority	The priority score of the patient
Status	The status of the patient
Day A	The nth day within the simulation environment that the patient receives a bed
Time A	The time of the day within the simulation environment that the patient receives a bed
Day D	The nth day within the simulation environment that the patient is discharged
Time D	The time of the day within the simulation environment that the patient is discharged
Time left in ER	The number of minutes the patient still has to stay in the ER
Total wait time	The total wait time of the patient
SF Score	The patient's satisfaction score

3. Satisfaction

Name	**Description**
Sim N	The seed number of the simulation
Day	The nth day within the simulation environment
Average SF per Day	The average satisfaction score of all patients by the end of the day