U-SORT: Utah Support for Emergency Operating Room Triage

Chengyu Yang
University of Utah

UUCS-24-010

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

3 December 2024

Abstract

Emergency Operating Room (EOR) scheduling in hospitals traditionally follows a firstcome, first-served (FCFS) approach. However, with advancements in artificial intelligence, there is potential to optimize this process. This study explores a priority-based scheduling method, U-SORT, aimed at minimizing surgery delays by prioritizing patients based on their predicted urgency. The ultimate goal is for U-SORT to build upon deep learning techniques used in Emergency Department triage, such as those proposed by Ivanov, to help predict the most appropriate treatment pathway, surgery time, and surgery duration time for patients [5]. This thesis evaluates whether U-SORT, a priority-based triage method, can reduce the number of delayed surgeries compared to the conventional FCFS method.

U-SORT: UTAH SUPPORT FOR EMERGENCY OPERATING ROOM TRIAGE

by

Chengyu Yang

A Senior Thesis Submitted to the Faculty of
The University of Utah
In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science

In

Computer Science

Approved:

Thomas C Henderson

Thomas C. Henderson
Thesis Faculty Supervisor

Mary Hall

Mary Hall
Director, School of Computing

Jim de St. Germain
Director of Undergraduate Studies

November 2024

ABSTRACT

Emergency Operating Room (EOR) scheduling in hospitals traditionally follows a first-come, first-served (FCFS) approach. However, with advancements in artificial intelligence, there is potential to optimize this process. This study explores a priority-based scheduling method, U-SORT, aimed at minimizing surgery delays by prioritizing patients based on their predicted urgency. The ultimate goal is for U-SORT to build upon deep learning techniques used in Emergency Department triage, such as those proposed by Ivanov, to help predict the most appropriate treatment pathway, surgery time, and surgery duration time for patients [5]. This thesis evaluates whether U-SORT; a priority-based triage method, can reduce the number of delayed surgeries compared to the conventional FCFS method.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Emergency room triage decision support systems have been developed for hospital emergency departments (ED) and have demonstrated the ability to improve patient outcomes and hospital resource utilization [2, 3, 5, 7, 8]. This work is part of a preliminary study to determine the a priori relative value of the current Emergency Operating Room (EOR) triage method: first-come-first-served (FCFS). Suppose it can be determined that alternative triage algorithms can outperform FCFS. In that case, that provides a good basis to do a more in-depth study on the application of deep learning to emergency operation room (EOR) triage in which a broader range of information can be brought to bear both for medical outcomes and more optimal resource allocation.

Many healthcare systems operate at full or near-full capacity, requiring triage decisions for valuable resources such as operating room access. The current system for managing urgent cases at the University of Utah is largely FCFS, with little attention paid to the disease process or its impact on healthcare resource utilization. Other centers function similarly or employ individuals who triage patients into urgency-based slots based on disease severity.

Figure 1.1 shows how hospitals typically organize EOR usage. As stated above, most hospitals schedule operations using FCFS, meaning a patient's scheduled surgery time will be the first available slot after their arrival. While FCFS makes scheduling patients in the EOR triage room extremely simple, it is unable to modify the patient triage plan time based on the patient's priority. It can result in an urgent patient being booked after other non-urgent patients. This implies that while non-urgent patients have more time to wait for their urgent operation to be completed, high-priority patients will resignedly have their surgery delayed. After patients are examined and data collected, they are assigned an urgency-based priority for an operation; each priority has an associated allowable delay to the start of surgery; e.g., there may be three priorities with scheduling slots S1: 0 to 3 hours, S2: 3 to 12 hours, and S3: 12 to 24 hours. This makes it possible to schedule less urgent patients later leaving the EOR available for more urgent cases until that time. Note that it is also possible that no urgent cases arrive in the meantime and the EOR goes unused.

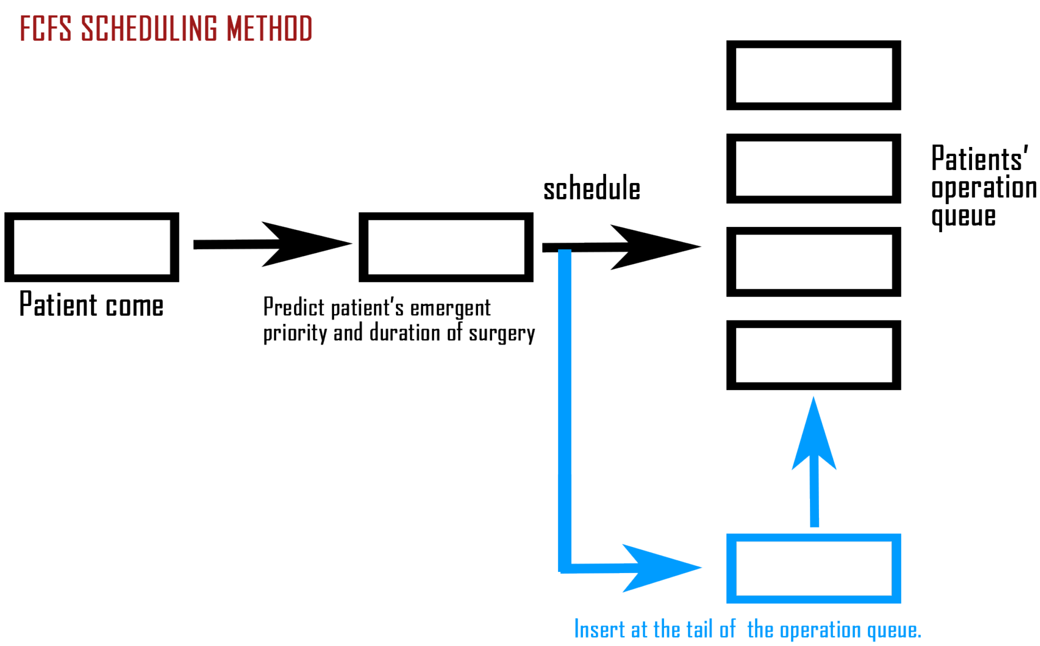

Figure 1.1. How hospitals schedule operating rooms in general (FCFS).

Figure 1.2 shows the idea behind our proposed U-SORT algorithm, which assigns patient operation times according to the urgency of the case. The exact time of surgery will be scheduled based on the patient's pathology. This allows the surgery time of patients whose pathological conditions are not critical to be postponed appropriately to free up time to deal with patients with more urgent conditions.

The hypothesis is that there are priority-based triage scheduling algorithms that will produce fewer delays than FCFS; this may depend on parameters of the specific scenario (e.g., patient arrival rates, mean surgery duration, number of scheduling slots, etc.).

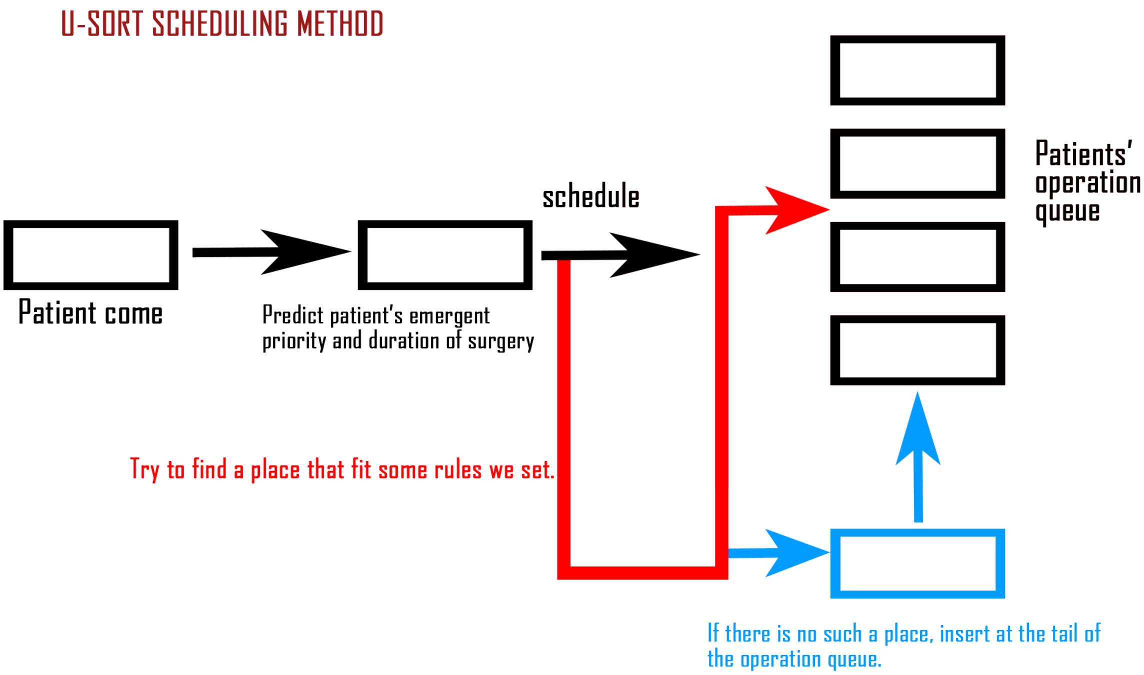

Figure 1.2. How U-SORT schedules operating room.

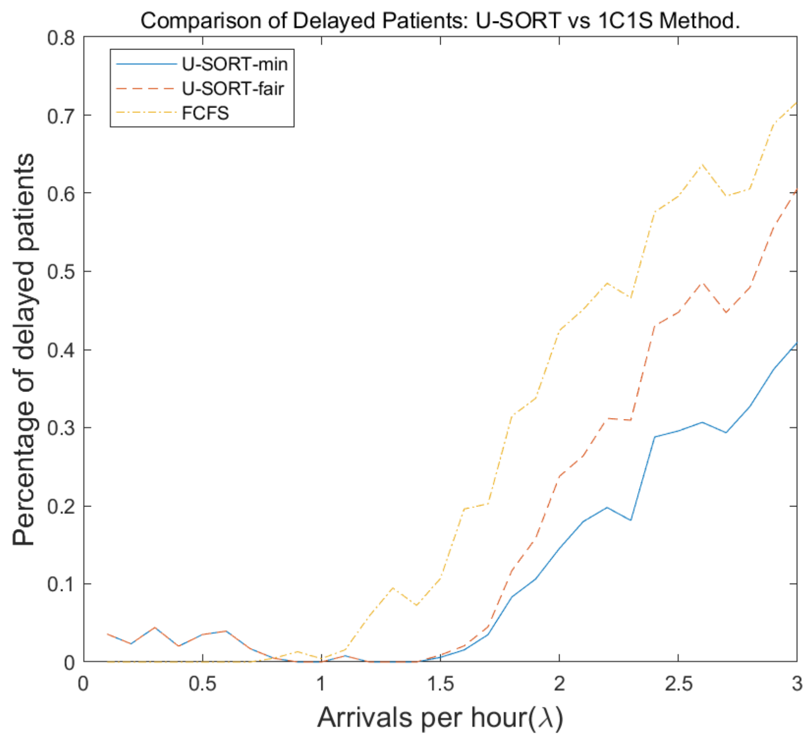

Figure 1.3. Comparison of FCFS and U-SORT in terms of the percentage of delayed patients. The simulation runs over 24 hours, with the mean surgery time of one hour and arrivals per hour ranging from 0.1 to 3 in steps of 0.1.

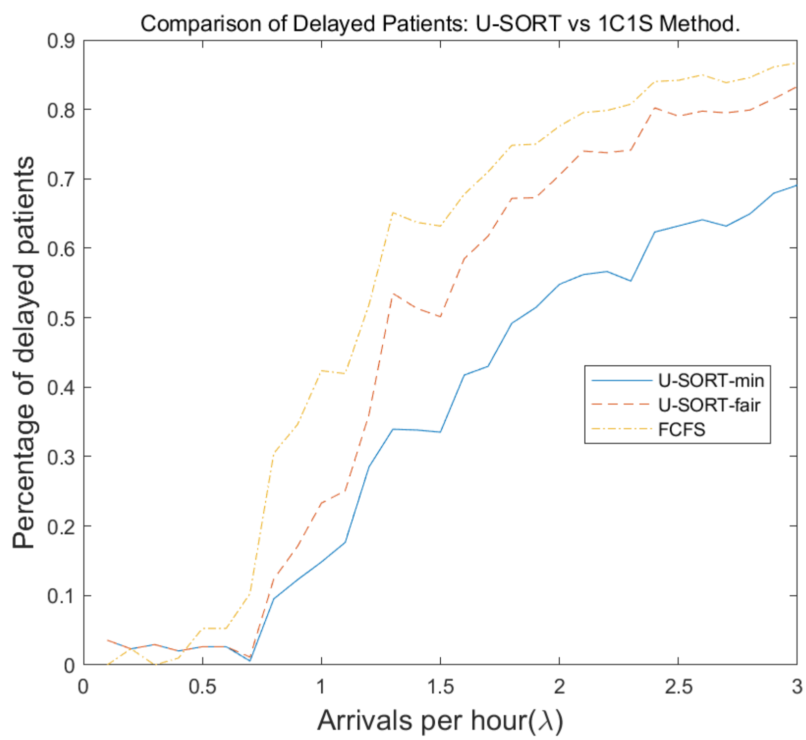

Figure 1.4. Comparison of FCFS and U-SORT in terms of the percentage of delayed patients. The simulation runs over 24 hours, with the mean surgery time of two hours and arrivals per hour ranging from 0.1 to 3 in steps of 0.1.

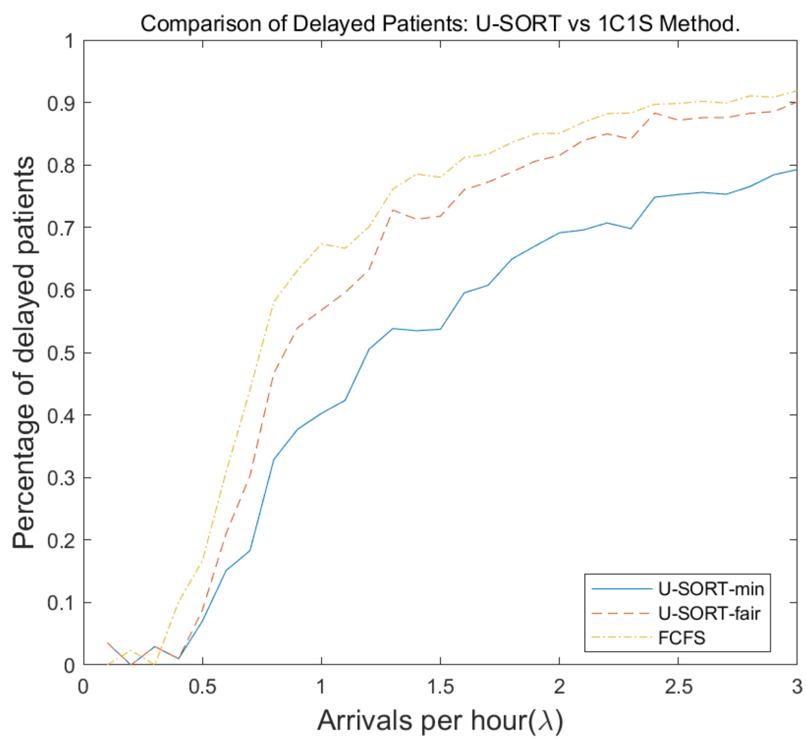

Figure 1.5. Comparison of FCFS and U-SORT in terms of the percentage of delayed patients. The simulation runs over 24 hours, with the mean surgery time of three hours and arrivals per hour ranging from 0.1 to 3 in steps of 0.1.

To simulate to test a patient going to a hospital EOR, we will run a method that randomly generates a set of patient arrival forms based on a few variables. There are 5 parameters, T, λ, P_{list}, d_{mu}, d_{var}, and d_{min}. T represents the total duration of running the simulation. For example, $T = 24$ means that we will continuously simulate the situation of patients going to the hospital within 24 hours. λ represents how many patients arrive at the EOR per hour. If one simulation has parameters $T = 24$ and $\lambda = 2$, the total number of patients will be around 48. P_{list} gives the probability of the patient's urgency priorities. In this project, we assume that there are 3 priorities in total and the probability that a patient is in priority 1, 2, and 3 is 0.05, 0.6, and 0.35 respectively. In addition, the on-time surgery time for the three priorities is 0 - 3 hours, 3 - 12 hours, and 12 - 24 hours after arrival, respectively. This means a patient is more urgent if he/she has lower priority. Finally, d_{mu}, d_{var}, and d_{min} are variables that control the generation of patients' surgery duration. Overall, patients' surgery time will be generated from the log-normal distribution defined by the input mean and variance, but the surgery time should not be less than d_{min}.

Figures 1.3 to 1.5 that fixed $T = 24$, $d_{var} = 0.1$, $P_{list} = [0.05; 0.6; 0.35]$, $d_{min} = 1$ show in 10 randomly generated patient sets what percentage of patients will be operated on delayed depending on λ in the range 0.1 to 3. Figure 1.3 fixed $d_{mu} = 1$, Figure 1.4 fixed $d_{mu} = 2$, and Figure 1.5 fixed $d_{mu} = 3$. The solid blue line shows the percentage of patient delays for U-SORT-min; the dashed orange line and the dot-dash yellow line represent that for U-SORT-fair and FCFS, respectively.

As is evident, definitely, the U-SORT algorithm performs better than FCFS. For instance, whatever d_{mu} is, during λ increasing, the two lines which represent U-SORT-min and U-SORT-fair patients percentage delay always lower. Thus, preliminary simulations show that U-SORT performs better than FCFS in situations where healthcare systems are operating at full or near-full capacity. This satisfies our hope that the new algorithm can allocate EOR Triage Room resources more efficiently and economically.

CHAPTER 2

BACKGROUND

Utah's priority-based triage system under development support for emergency operating room triage, U-SORT, is based on two main ideas, an urgency priority assignment method and job scheduling methods. One of them is how we separate different patients according to the urgency of their condition, physical condition, drug resistance, and other external and internal factors that can affect the patients' surgery. Chen et al. [2] developed a deep neural network (DNN) to accurately predict emergency department patients' treatment and reported better results than the Rapid Emergency Medicine Score. This is effective in alleviating overcrowding in the Emergency Department. Gebrael [4] describes a retrospective study that analyzes patient triage in the Emergency Department and compares the performance of ChatGPT with that of emergency room physicians on triage decisions, diagnostic accuracy, treatment recommendations, and emergency severity index score prediction. Ivanov [5] investigated the accuracy and impact of the Emergency Severity Index (ESI) triage system; the study found that approximately one-third of encounters were misdiagnosed, with 3.3% underdiagnosed and 28.9% overdiagnosed. Finally, and perhaps most importantly, Ivanov proposed an attention-based convolutional neural network to assess medical urgency based on the patient's condition and to recommend the most appropriate point of care and treatment time. These studies support the idea that a deep-learning network model may be suitable for inferring patient pathology and scheduling in the EOR.

When the priority of jobs cannot be determined, First-Come, First-Served (FCFS) is the most straightforward scheduling method, prioritizing jobs based on their arrival order, serving earlier jobs before later ones. Another widely used method is Shortest Job First (SJF), which sorts processes by their arrival time and selects the one with the shortest burst time for execution. A more complex approach is the Highest Response Ratio Next (HRRN), where each job's priority is dynamically calculated using its response ratio, improving fairness by balancing waiting time and burst time. The response ratio is computed as:

$$\textbf{response ratio} = 1 + \frac{\text{waiting time of a process so far}}{\text{estimated run time}}. \quad (2.1)$$

According to research by Wong and Kim [6, 11], most institutions adopt a First-Come, First-Served (FCFS) protocol for patient management. However, with advances in deep learning that enable the classification of patient urgency, scheduling methods in the EOR can shift towards Priority Scheduling, as exemplified by the U-SORT model. U-SORT aims to reduce the number of delayed patients by adjusting FCFS to prioritize urgent cases. In the following chapter, we will define U-SORT and compare its performance with FCFS, focusing on patient delay metrics.

Finally, one of the most important things is the background of how we run our simulation. We use DES (Discrete Event Simulation) as described in [1]. DES can simulate events that occur in the system, and the system state changes when the event occurs. The simulation process usually relies on time advancement, updating the system state when the event occurs, which indicates it is good for simulating situations that depend on time. In order to run our simulation, there are some non-deterministic aspects and other parameters. The following table lists the variables used in the simulation in detail.

Variable	**Description**
λ	Arrival rate (Average number of patient arrivals per hour)
P_{list}	Discrete probability set for patients' priority
d_{mu}	Mean value for patients' surgery duration distribution
d_{var}	Variance for patients' surgery duration distribution
d_{min}	Minimum for patients' surgery duration
$slot$	The optimal surgery start interval corresponds to each patient's priority level
T	Total simulation time

Table 2.1. Description of simulation variables to be assigned.

When simulating the patient arrival process in an emergency medical setting, a common way is using the Poisson Process:

$$a_i(\text{The } i^{\text{th}} \text{ patient arrival time}) \sim PD(\lambda, T) \quad (2.2)$$

described by Ross [9]. This is because the Poisson process can well describe the occurrence of random events, where the time intervals between events are independent. First of all, we initially set parameters λ, T. Then we can generate the i^{th} patient's arrival time a_i, sampling a random number U from a uniform distribution, restricting it to the interval $(0, 1)$, and then applying

$$a_0 = 0; \tag{2.3}$$

$$a_i = a_{i-1} - \frac{\log(U)}{\lambda}. \tag{2.4}$$

When output value $a_i \geq T$, the simulation will stop generating the next patient's arrival time, otherwise, the simulation will continue. After all patients' arrival times are generated, the simulation will input the arrival list to generate every patient's priority by using a discrete probability set P_{list}, where P_i represents the probability of each patient being assigned this priority.

$$P_{list} = \{P_1, P_2, ..., P_n\} \tag{2.5}$$

After assigning all patients a priority, the last thing is to generate every patient's surgery duration time from a log-normal distribution. Log-normal distribution is a continuous probability distribution that applies to situations where the logarithm of a random variable follows a normal distribution. A log-normally distributed random variable can only take positive values and right-skewed. That is, in most cases, the left side of the distribution is steeper and the right side is flatter, with a long tail effect. In research posted by Strum [10], they have proven that log-normal distribution is better than normal distribution when simulating surgery duration time. We will use given d_{mu}, d_{var} to calculate μ_{log}, and σ_{log} for log-normal distribution as follows:

$$\mu_{log} = \log(\frac{d_{mu}^2}{\sqrt{d_{var} + d_{mu}^2}}) \tag{2.6}$$

$$\sigma_{log} = \sqrt{\log(1 + \frac{d_{var}}{d_{mu}^2})}. \tag{2.7}$$

With μ_{log}, and σ_{log} we calculate, we can generate patients' duration time from the log-normal distribution with $\mu = \mu_{log}$ and $\sigma^2 = \sigma^2_{log}$. Finally, check and replace every patient's duration time d_i to d_{min} when $d_i < d_{min}$.

CHAPTER 3

METHOD

3.1 Algorithm

This section defines three EOR scheduling methods: FCFS, U-SORT-min, and U-SORT-fair.

3.1.1 FCFS

We used two methods to implement the FCFS method as the baseline method, direct implementation and linear programming method. As a direct implementation of FCFS, when the i^{th} patient arrives ($i \geq 1$), FCFS will schedule the patient's surgery time based on the following conditions:

1. Perform surgery on this patient immediately (no patient is currently undergoing surgery);

2. Schedule the patient's surgery after the $i-1^{\text{th}}$ patient's surgery is completed (otherwise);

Variable	description
A	Coefficient matrix for linear programming
n	Patients number in giving patient list
a	Vector of arrival times for all patients in the list.
d	Vector of surgery duration time for all patients in the list except the last patient
b	Result matrix for linear programming
X	Vector of patient surgery start times scheduled by FCFS
a_i	The arrival time of the i^{th} patient
x_i	The surgery start time of the i^{th} patient
d_i	The surgery duration time of the i^{th} patient

Table 3.1. Variables used in linear programming and their descriptions.

In the implementation of linear programming, giving a random patient list, we assume as Table 3.1. Set the objective function as

$$\min \sum_{i=1}^{n} x_i \tag{3.1}$$

To ensure the FCFS scheduling requirement, add the following constraints: For each patient i, the start time x_i of the operation must be greater than or equal to the arrival time a_i, that is,

$$x_i \geq a_i, \quad \forall i. \tag{3.2}$$

In addition, for adjacent patients i and $i+1$, the operation start time x_{i+1} of $i+1$ must not be earlier than the operation end time $x_i + d_i$ of i, that is,

$$x_{i+1} \geq x_i + d_i. \tag{3.3}$$

Then, combine Equations 3.2 and 3.3

$$-x_i \leq -a_i \tag{3.4}$$

$$x_i - x_{i+1} \leq -d_i \tag{3.5}$$

That is

$$\begin{bmatrix} -1, 0 \\ 0, -1 \\ 1, -1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ x_{i+1} \end{bmatrix} \leq \begin{bmatrix} -a_i \\ -a_{i+1} \\ -d_i \end{bmatrix} \tag{3.6}$$

We apply Equation 3.6 to all $x_i \in X$ we get

$$A \cdot X \leq b \tag{3.7}$$

Where

$$A_1 \text{ is an } (n-1) \times n \text{ matrix,} \tag{3.8}$$

$$\text{the } k^{\text{th}} \text{ element of the } k^{\text{th}} \text{ row is 1, and the } k+1^{\text{th}} \text{ element is -1.} \tag{3.9}$$

$$A = \begin{bmatrix} -I \\ A_1 \end{bmatrix}, \tag{3.10}$$

$$b = \begin{bmatrix} -a \\ -d \end{bmatrix}. \tag{3.11}$$

Finally, we use the *linprog* function in MATLAB to solve the problem and obtain an optimal patient surgery start time vector X that satisfies FCFS scheduling.

3.1.2 U-SORT-fair

When the first patient arrives, U-SORT-fair, like the FCFS method, will place him in the first vacant slot that can be operated on, that is, the surgery will be performed immediately when the patient arrives. However, when the i^{th} patient arrives ($i \geq 2$), the U-SORT-fair method provides a method to compare it with the $i-1^{\text{th}}$ patient undergoing surgery to determine whether the surgery sequence needs to be adjusted. The adjustment method of U-SORT-fair is based on the following conditions:

1. The emergency priority of the i^{th} patient is higher than that of the $i-1^{\text{th}}$ patient undergoing surgery;
2. The surgery start time of the $i-1^{\text{th}}$ patient undergoing surgery is later than the arrival time of the i^{th} patient;
3. Scheduling the surgery of the i^{th} patient before the $i-1^{\text{th}}$ patient undergoing surgery will not cause the surgery of the $i-1^{\text{th}}$ patient to be delayed outside of its priority slot.

If and only if all of the above conditions are met, the U-SORT-fair method will schedule the surgery of the i^{th} patient before the $i-1^{\text{th}}$ patient undergoing surgery, and continue to compare the i^{th} patient with the $i-2^{\text{th}}$ patient undergoing surgery. This process continues until at least one of the above conditions is not met, or the surgery of the i^{th} patient is scheduled before the 1^{st} patient waiting to undergo surgery. After determining the surgical insertion position of the i^{th} arriving patient, U-SORT-fair will run a judgment method to detect whether the surgery of the i^{th} patient is delayed based on the patient's urgent priority p_i of the i^{th} patient and the optimal surgery start interval of that urgent priority. If the surgery is judged to be delayed, the surgery of the i^{th} patient will be scheduled at the end of the current surgery queue to ensure that the surgeries of other patients are not affected. In order to minimize delayed patient numbers, the impact of patients whose surgery has been inevitably delayed on patients who have not yet been scheduled for surgery must be reduced.

3.1.3 U-SORT-min

U-SORT-min is similar to U-SORT-fair but has added a new constraint.

1. If $i-1^{\text{th}}$ patient has been inevitably delayed, set i^{th} patient's surgery before $i-1^{\text{th}}$ patient

3.2 Simulation Framework

We use FCFS and U-SORT methods to simulate the EOR surgery schedule when the surgery is performed 24 hours a day. Assume $d_{i,USORT-min} \in D_{USORT-min}$, $d_{i,USORT-fair} \in D_{USORT-fair}$ represents the average number of patients delayed caused by U-SORT-min and U-SORT-fair under the i^{th} set of variables. $d_{i,FCFS}$ represents the average number of patients delayed caused by FCFS under the i^{th} set of variables.

$$d_{i,USORT-min} = \frac{\text{number of patients delayed by U-SORT-min under the } i^{\text{th}} \text{ variable set}}{N_{trials}} \tag{3.12}$$

$$d_{i,USORT-fair} = \frac{\text{number of patients delayed by U-SORT-fair under the } i^{\text{th}} \text{ variable set}}{N_{trials}} \tag{3.13}$$

$$d_{i,FCFS} = \frac{\text{number of patients delayed by FCFS under the } i^{\text{th}} \text{ variable set}}{N_{trials}} \tag{3.14}$$

Finally, U-SORT and the baseline method FCFS were compared by using the mean number of patient delays for U-SORT and FCFS under specific group variables, the Mann–Whitney U test, and the calculation of improvement rates.

For N_{set} variable sets, U-SORT-min, U-SORT-fair, and FCFS will generate a set of data sets $D_{USORT-min}$, $D_{USORT-fair}$ and D_{FCFS}, which represent the average number of delayed patients simulated N_T times. For any two result sets, we use a paired sample t-test to help test the significance. For example, we choose $D_{USORT-min}$ and D_{FCFS}, the Mann–Whitney U test is:

- n_1 and n_2 are the sizes of two samples respectively.
- R_1 is the rank sum of the first sample.

calculate

$$U = n_1 \times n_2 + \frac{n_1 \times (n_1 + 1)}{2} - R_1 \tag{3.15}$$

After calculating the U value, we can find the p-value through statistical software. If the p-value is less than the significance level such as 0.05, the null hypothesis can be rejected and the distribution of the two samples is considered to be significantly different.

We also calculate the average improvement rate of $D_{USORT-min}$ to D_{FCFS} to quantify the improvement of U-SORT-min compared to FCFS.

$$\text{improvement rate} = \frac{\overline{D_{FCFS}} - \overline{D_{USORT-min}}}{\overline{D_{FCFS}}} \times 100\% \tag{3.16}$$

Variable	**description**
A	Coefficient matrix for linear programming
X_{FCFS}	Vector of patient surgery start times scheduled by FCFS for one patient list
i	$1 \leq i \leq N_{set}$
a_i	The arrival time of the i^{th} patient scheduled by FCFS for one patient list
x_i	The surgery start time of the i^{th} patient scheduled by FCFS for one patient list
d_i	The surgery duration time of the i^{th} patient scheduled by FCFS for one patient list
N_{trials}	For a set of variables, the number of trials to run the experiment with different seeds
N_{λ}	The number of different λ values that can be chosen in the experiment
N_{mu}	The number of different $d_m u$ values that can be chosen in the experiment
N_{var}	The number of different $d_v ar$ values that can be chosen in the experiment
N_T	The number of different T values that can be chosen in the experiment
N_{set}	Number of variable sets
$D_{USORT-min}$	Vector of Average number of patients delayed caused by U-SORT under N_{set} sets of variables based on minimizing the number of delayed patients
$D_{USORT-fair}$	Vector of Average number of patients delayed caused by U-SORT under N_{set} sets of variables based on fairness
D_{FCFS}	Vector of Average number of patients delayed caused by FCFS under N_{set} sets of variables

Table 3.2. Variables used in the method and their descriptions.

CHAPTER 4

EXPERIMENTS

4.1 Experimental Setup

We fix the range and step size of the required variables as shown in Table 4.1. That is, we will have 15,840 sets of variables. For every set of parameters, we use it and set a different random seed to generate a total of 10 patient lists. After we get all 158,400 patient lists, we run the U-SORT-min, U-SORT-fiar, and FCFS methods to schedule EOR with it and calculate overall delay patients. Finally, after all three methods are completed, the output will be a matrix with a shape of 15,840 $\times$ 4. In this matrix, the first column represents the total number of delayed patients using U-SORT-min in $N_{trials} = 10$ trials, while the second and third columns represent the numbers using U-SORT-fair and FCFS, respectively. The fourth column represents the total arrival of patients through 10 trials. We apply the Mann–Whitney U test to the pairwise result sets and calculate the improvement rate. The results are shown in Table 4.2, where p represents the p-value of

Variable	Interval	Step
λ	[0.1, 3]	0.1
P_{list}	[0.05; 0.6; 0.35]	0.05 priority 1; 0.6 priority 2; 0.35 priority 3
d_{mu}	[0.5, 3]	0.5 hour
d_{var}	[0, 1]	0.1
d_{min}	0.5	- hour
$slot$	[0, 3; 3, 12; 12, 24]	0-3 priority 1; 3-12 priority 2; 12-24 priority 3
T	[24, 192]	24 hours
N_{trials}	10	-
N_{λ}	30	-
N_{mu}	6	-
N_{var}	11	-
N_T	8	-
N_{set}	$30 \times 6 \times 11 \times 8 = 15840$	-

Table 4.1. Variables' value interval and step.

the Mann–Whitney U test between the pairwise sets, and I represents the improvement rate of the first set to the second set between the pairwise sets.

4.2 Experimental Results

To simplify the statement, we use $D_{USORT-min}$, $D_{USORT-fair}$, and D_{FCFS} instead of the first, second, and third columns of the output matrix. First, while we get the output, we test whether U-SORT-fair and U-SORT-min have significant differences with FCFS. We run three Mann-Whitney U Tests and then calculate the improved rate. First, we will test three result set populations; results are shown in Table 4.3.

After that, we fixed the value of T to 24 and d_{var} to 0 to simulate the working of the three scheduling methods when λ increases from 0.1 to 3 and d_{mu} increases from 0.5 to 3. We stress-test the three methods by increasing the values of λ and d_{mu} and test the improvement of the two U-SORT methods compared with FCFS under overloaded hospital operation. Then, as shown in Figure 4.1, the red plane represents the average number of delayed patients generated by the U-SORT-min scheduling method, the green plane represents the average number of delayed patients generated by the U-SORT-fair scheduling method, and the blue plane represents the average number of delayed patients generated by the FCFS scheduling method. As can be seen from Figure 4.1, under high

Variable	**descripting**
$p_{class,firstSet,secondSet}$	P-value of the Mann–Whitney U test between the first result set and the second result set.
$I_{class,firstSet,secondSet}$	Improvement rate between the first result set and the second result set.

Table 4.2. Description of variables for the result of Mann–Whitney U test and Improvement rate calculation.

Variable	**Value**
$p_{overall,U-SORT-min,U-SORT-fair}$	0.0000
$I_{overall,U-SORT-min,U-SORT-fair}$	32.1207
$p_{overall,U-SORT-min,FCFS}$	0.0000
$I_{overall,U-SORT-min,FCFS}$	34.0932
$p_{overall,U-SORT-fair,FCFS}$	0.6811
$I_{overall,U-SORT-fair,FCFS}$	2.9060

Table 4.3. P-value and improvement rate for three methods run in 15,840 variable sets.

load operation, the performance of the three methods all dropped rapidly, but the performance of U-SORT-fair dropped more than that of U-SORT-min so that when $\lambda = 3$, $d_{mu} = 3$, the number of delayed patients caused by it was close to the FCFS scheduling method.

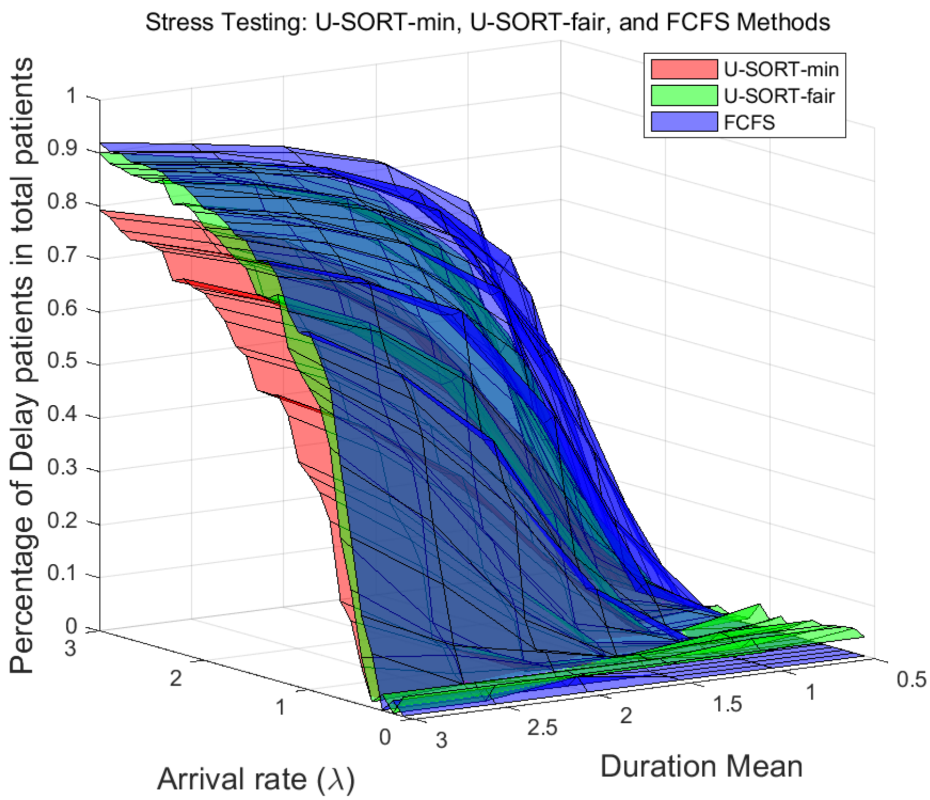

Figure 4.1. The percentage of patients delayed in total arrival patients produced by the three scheduling methods when d_{mu} increases from 0.5 to 3 and λ increase from 0.1 to 3 but $T = 24, d_{var} = 0$.

Variable	Value
$p_{stress,U-SORT-min,U-SORT-fair}$	0.2079
$I_{stress,U-SORT-min,U-SORT-fair}$	22.2636
$p_{stress,U-SORT-min,FCFS}$	0.0575
$I_{stress,U-SORT-min,FCFS}$	31.8799
$p_{stress,U-SORT-fair,FCFS}$	0.5924
$I_{stress,U-SORT-fair,FCFS}$	12.3704

Table 4.4. P-value and improvement rate for three methods when d_{mu} increases from 0.5 to 3 and λ increases from 0.1 to 3 but $T = 24, d_{var} = 0$.

Then, we fix all variables except T, making $\lambda = 1.2$, $d_{mu} = 1$, $d_{var} = 0$, and find the ability of the three scheduling methods to maintain performance in long-term operation as time increases. Because when T increases, the total number of patients also increases. It is meaningless to compare only the number of delayed patients. Therefore, we use the percentage of delayed patients to the total number to evaluate the performance of each method in the current environment. As shown in Figure 4.2, we will also calculate the p-value for the Mann-Whitney U Test and relative improvement rate.

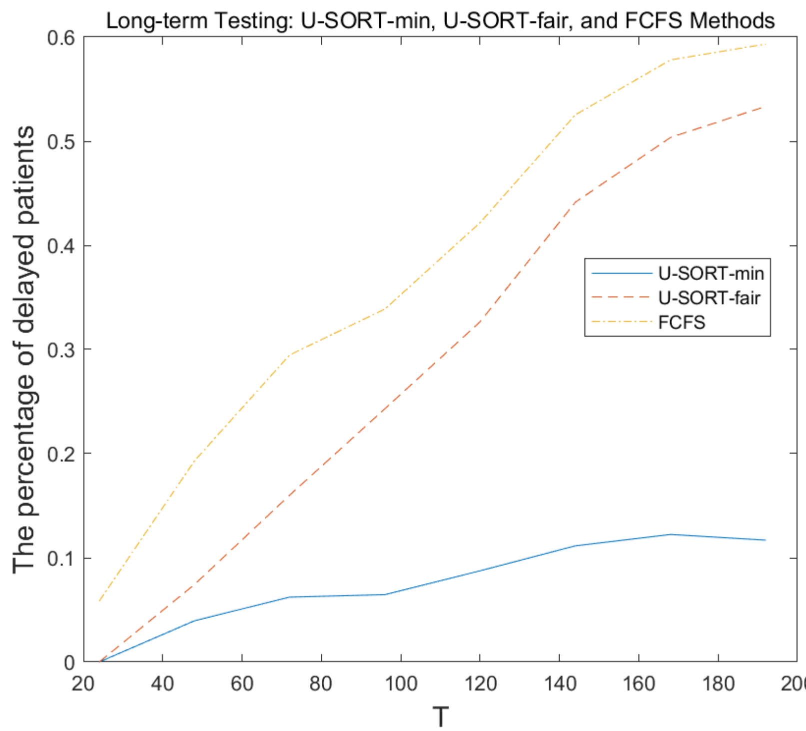

Figure 4.2. Long-term performance of the three methods for fixed $\lambda = 1.2$, $d_{mu} = 1$, and $d_{var} = 0$.

In addition, we fixed the parameters $\lambda = 1$, $T = 24$, and $d_{mu} to 2$, and only adjusted d_{var} to generate more unstable patient surgery times, so as to test the stability of the three methods when the surgery time fluctuates greatly. As shown in Figure 4.3, the surgery duration variance control variable d_{var} starts from 0 and increases to 1 in steps of 0.1. Using the patient list generation method introduced in the previous chapter, we generate 10 patient lists for each group of variables. In this test, for each group of variables, the 10 patient lists are generated, the three scheduling methods are run, the number of delayed patients for each method is calculated, and then the variance of the number of delayed patients for 10 simulations under this group of variables is calculated. This means that when d_{var} increases, each method will get 11 variance values, so as to compare the stability of each method.

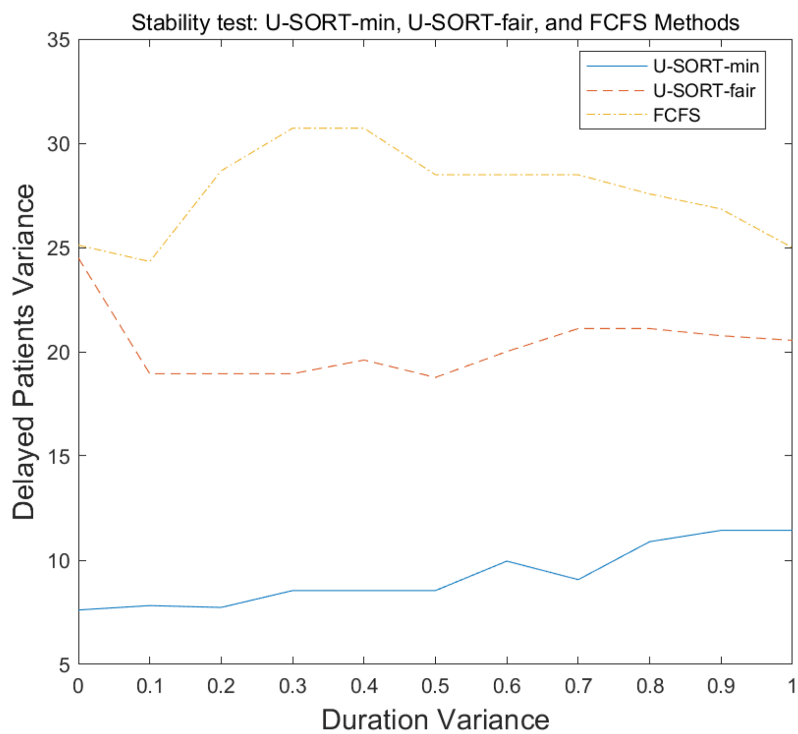

Figure 4.3. Stability of the three methods when $\lambda = 1$, $T = 24$, and $d_{mu} = 2$; d_{var} starts from 0 and increases to 1 in steps of 0.1.

Finally, we want to find in what situation, U-SORT-min is significantly better than U-

Variable	**Value**
$p_{long,U-SORT-min,U-SORT-fair}$	0.0289
$I_{long,U-SORT-min,U-SORT-fair}$	73.4974
$p_{long,U-SORT-min,FCFS}$	0.0047
$I_{long,U-SORT-min,FCFS}$	79.8571
$p_{long,U-SORT-fair,FCFS}$	0.3823
$I_{long,U-SORT-fair,FCFS}$	23.9966

Table 4.5. P-value and improvement rate for three methods when fixed $\lambda = 1.2$, $d_{mu} = 1$, and $d_{var} = 0$ but T increase from 24 to 192 with step is 24 hours.

SORT-fair. That is, we screen out under what variable conditions are inclined to increase the difference of patients delayed between U-SORT-min and U-SORT-fair. To achieve this goal, we separate the overall result set into two different results, the Non-significant set and the Significant set, according to the difference in delayed patients number. When the difference in percentage of delayed patients in total arrival patients is less than 1%, the set of variables will be added to the Non-significant set, otherwise added to the Significant set. The final results are shown in Figures 4.4 and 4.5. Subsequently, the Mann-Whitney U Test is performed on the significant and insignificant cases and the improvement rate is calculated.

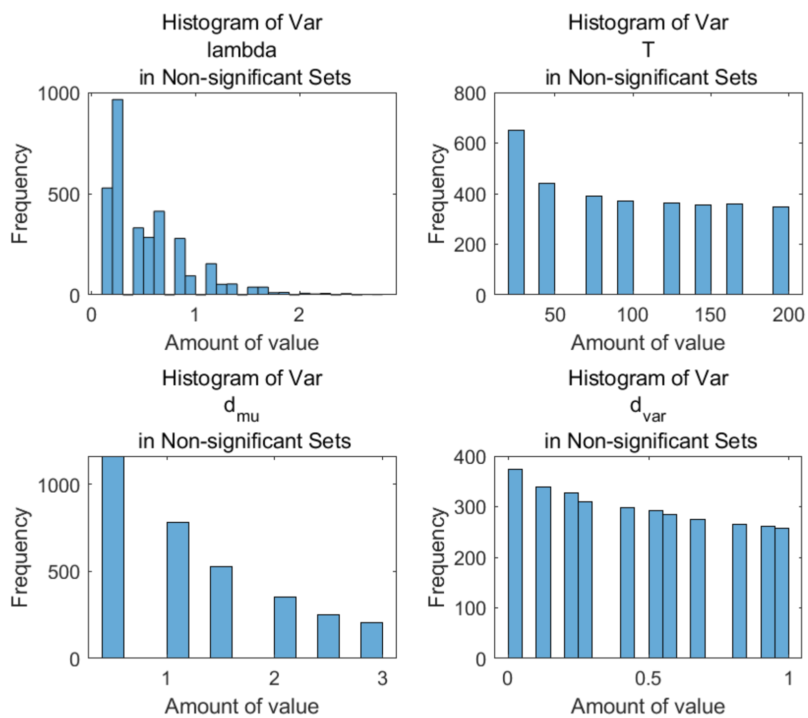

Figure 4.4. The variables' distribution pattern of variables in the non-significant set.

Combining Figures 4.4, 4.5, and Table 4.6 shows that the distribution of variables T and d_{var} values is relatively even in the significant or non-significant set. Moreover, running the Mann-Whitney U Test on the non-significant sets of U-SORT-min and U-SORT-fair resulted in $p_{non-significant,U-SORT-min,U-SORT-fair} = 0.1546$, indicating that in the non-significant set, the performance comparison of U-SORT-min and U-SORT-fair is not much different,

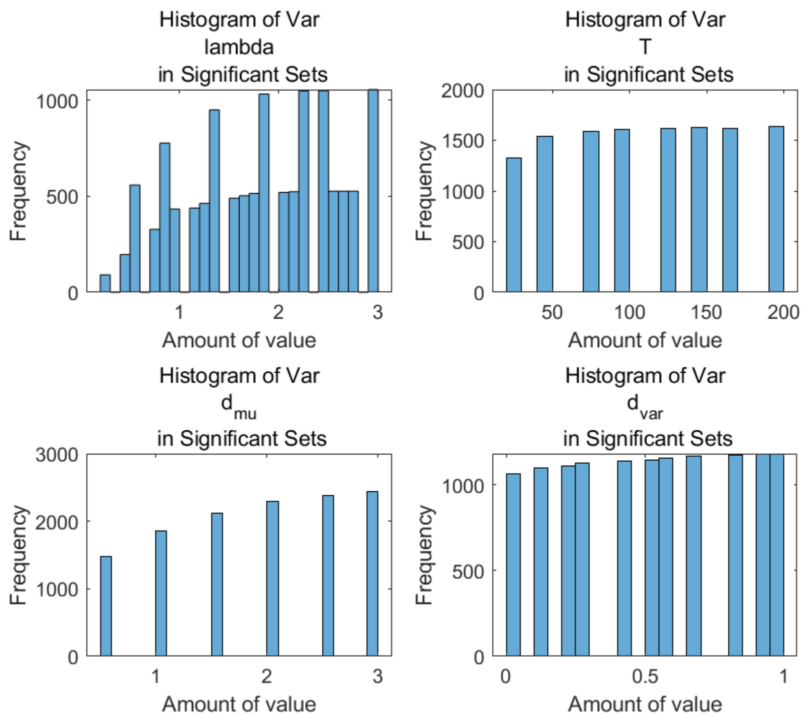

Figure 4.5. The distribution pattern of variables in the significant set.

and U-SORT-min has only a 4% improvement rate. This means these two variables have little impact on whether U-SORT-min is significantly better than U-SORT-fair. On the contrary, although there are some exceptions, the amount of occurrences of λ and d_{mu} in the non-significant set generally decreases as its value increases, while in the significant set, the number of occurrences gradually increases as its value increases. increase. This means that λ and d_{mu} are the two main variables that affect the performance difference between U-SORT-min and U-SORT-fair. When the values of these two variables increase, U-SORT-min becomes more likely to be significantly better than U-SORT-fair.

Variable	**Value**
$p_{non-significant,U-SORT-min,U-SORT-fair}$	0.1546
$I_{non-significant,U-SORT-min,U-SORT-fair}$	4.4976
$p_{significant,U-SORT-min,U-SORT-fair}$	0.0000
$I_{significant,U-SORT-min,U-SORT-fair}$	32.1601

Table 4.6. P-value and improvement rate for the Non-significant set and the Significant set.

CHAPTER 5

CONCLUSIONS

This thesis explores the potential of the U-SORT scheduling algorithm to outperform traditional FCFS in reducing surgical delays. By introducing U-SORT-min and U-SORT-fair, this study aims to provide a more efficient resource allocation strategy for emergency operating rooms. In all simulations, U-SORT-min has a 32% improvement rate compared to U-SORT-fair and has a 34% improvement rate for FCFS. The results clearly show that U-SORT-min outperforms U-SORT-fair and FCFS in reducing delayed surgery. The stress test actually simulates the performance comparison of the three scheduling methods U-SORT-min, U-SORT-fair, and FCFS within 24 hours when the number of patients arriving per hour, i.e., λ, and the average duration of the patient's surgery, i.e., d_{mu}, increase to a given maximum value according to a given step value. Because the variance of the distribution of the duration of the patient's surgery is 0, i.e., $d_{var} = 0$, the duration of the surgery for each patient will be equal to d_{mu}. Because λ and d_{mu} increase at the same time according to a given step size, the stress test simulates the situation of the hospital EOR operating from low load to overload operation and gives a comparison of the work efficiency of the three scheduling methods in this case. While in the stress test, $p_{stress,U-SORT-min,FCFS} = 0.0575$ and $p_{stress,U-SORT-fair,FCFS} = 0.5924$. These two numbers indicate that when running Mann-Whitney U When tested, U-SORT-min and U-SORT-fair were not significantly different from FCFS. However, by comparing the improvement rate, U-SORT-min has an improvement rate of 31.8799% compared to FCFS in the stress test, while U-SORT-fair has an improvement rate of 12.3704% compared to FCFS. Both methods are better than the FCFS scheduling method and have at least a 10% improvement rate. In order to understand the long-term performance benefits of implementing U-SORT-min and U-SORT-fair scheduling methods compared to the FCFS scheduling method. After setting a series of variables such as the average duration of patient surgery to a reasonable value, the simulation time is only allowed to increase continuously to 192 hours, or 8 days, with a step size of 24 hours. The results are $I_{long,U-SORT-min,FCFS} = 79.8571$, $I_{long,U-SORT-fair,FCFS} = 23.9966$, which means that under this variable condition, U-SORT-min will have a 79% improvement rate compared to the FCFS scheduling algorithm. In addition, the performance of U-SORT-fair will also be 24% better than the FCFS scheduling method. Thus, through long-term testing, long-term operation of U-SORT-fair and U-SORT-min scheduling methods will greatly improve the resource allocation of the FCFS scheduling method and reduce the number of delayed patients. Finally, through the stability test, both U-SORT-min and U-SORT-fair performed better than the FCFS scheduling method. These findings highlight the importance of urgency scheduling in optimizing EOR utilization. U-SORT-min provides a practical solution for hospitals to effectively manage high-priority cases, especially in resource-constrained settings. Although the results are encouraging, the study is limited by its reliance on simulated patient data and predefined variable ranges. Real-world validation is needed to generalize the findings to different healthcare systems. In addition, because the simulation is limited to a 24-hour working day, future work can be implemented to adapt the hospital EOR working time system to be in line with the actual situation by shortening working hours.

REFERENCES

[1] E. Babulak and M. Wang, *Discrete event simulation*, Aitor Goti (Hg.): Discrete Event Simulations. Rijeka, Kroatien: Sciyo, (2010), p. 1.

[2] C.-H. Chen, J.-G. Hsieh, S.-L. Cheng, Y.-L. Lin, P.-H. Lin, and J.-H. Jeng, *Emergency department disposition prediction using a deep neural network with integrated clinical narratives and structured data*, International Journal of Medical Informatics, 139 (2020), pp. 104–146.

[3] F. Gao, B. Boukebous, M. Pozzar, E. Alaoui, B. Sano, and S. Bayat-Makoei, *Predictive models for emergency department triage using machine learning: A systematic review*, Obstetrics and Gynecology Research, 05 (2022).

[4] G. Gebrael, K. K. Sahu, B. Chigarira, N. Tripathi, V. Mathew Thomas, N. Sayegh, B. L. Maughan, N. Agarwal, U. Swami, and H. Li, *Enhancing triage efficiency and accuracy in emergency rooms for patients with metastatic prostate cancer: A retrospective analysis of artificial intelligence-assisted triage using chatgpt 4.0*, Cancers, 15 (2023).

[5] O. Ivanov, L. Wolf, D. Brecher, E. Lewis, K. Masek, K. Montgomery, Y. Andrieiev, M. McLaughlin, S. Liu, R. Dunne, K. Klauer, and C. Reilly, *Improving ed emergency severity index acuity assignment using machine learning and clinical natural language processing*, Journal of Emergency Nursing, 47 (2021), pp. 265–278.e7.

[6] J.-K. Kim, *Enhancing patient flow in emergency departments: A machine learning and simulation-based resource scheduling approach*, Applied Sciences, 14 (2024).

[7] B. Mueller, T. Kinoshita, A. Peebles, M. A. Graber, and S. Lee, *Artificial intelligence and machine learning in emergency medicine: a narrative review*, Acute medicine & surgery, 9 (2022), p. e740.

[8] Y. Raita, T. Goto, M. K. Faridi, D. F. Brown, C. A. Camargo, and K. Hasegawa, *Emergency department triage prediction of clinical outcomes using machine learning models*, Critical care, 23 (2019), pp. 1–13.

[9] S. Ross, *Simulation*, Statistical modeling and decision science, Academic Press, 1997.

[10] D. Strum, J. May, and L. Vargas, *Modeling the Uncertainty of Surgical Procedure Times: Comparison of Log-normal and Normal Models*, Anesthesiology, 92 (2000), pp. 1160–1167.

[11] T. Wong, A. G. Quinlan, E. Sarraf, B. H. Irwin, A. C. Stanley, H. T. Do, D. C. Novak, and M. H. Tsai, *The operational ramifications of a first-in first-out bump policy*, Perioperative Care and Operating Room Management, 30 (2023), p. 100302.

Name of Candidate:	Chengyu Yang
Birth date:	January 2 2001
Birth place:	Kunming, Yunnan
Address:	Salt Lake City, Utah, 84102