Optimization of Hospital Emergency Department

by

Mackenzie Robert Andrew Simpson
Lakehead University

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTERS

in the Department of Computer Science

Lakehead University

Optimization of Hospital Emergency Department

by

Mackenzie Robert Andrew Simpson
Lakehead University

Supervisory Committee

Dr. Salimur Choudhury, Supervisor
(Department of Computer Science, Lakehead University, Canada)

Dr. Dr. David Savage, Supervisor
(Northern Ontario School of Medicine, Canada)

Dr. Yimin Yang, Departmental Member
(Department of Computer Science, Lakehead University, Canada)

Dr. Yassine, External Member
(Department of Software Engineering, Lakehead University, Canada)

ABSTRACT

This thesis is centered around the topic of emergency department(ED) optimization. Working in conjunction with the Thunder Bay Regional Health Sciences Centre a simulation model was developed to determine an optimal physician schedule for the high acuity portion of the ED. The simulation uses patients generated based on the data provided. The simulation accounts for resource usage and coordinating physician patient interaction. As a secondary component to the thesis the minimum cut problem is investigated, as it has potential in aiding physicians in the ED. During this investigation a local search algorithm is proposed and the effects of parallelization are investigated.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to acknowledge the following individuals and institutions for their contributions to this thesis.

First and foremost I would like to thank my two advisors **Dr. Salimur Choudhury** and **Dr. Dr. David Savage**. They have both provided a great amount of council and support throughout my thesis.

Dr. Yimin Yang for his time and participation as a member of my supervisory committee and his helpful suggestions to improve my thesis.

Dr. Yassine for his time and participation as a member of my supervisory committee and his helpful suggestions to improve my thesis.

Last but not least I would like to thank **Faculty of Graduate Studies, Faculty of Science and Environmental studies & the National Science and Engineering Research Council** for their generous financial contributions to my education. Additionally the **Ontario Student Assistance Program** for their offer of the Ontario Graduate Scholarship.

DEDICATION

"Little by little, one travels far."

– J.R.R. Tolkien

I would like to dedicate this thesis to,

My parents who have financed a great part of my education and supported me throughout the process. In addition to this support in later years, they helped lay a foundation to put me on the course to a higher education from an early age.

My professors who have had a great deal of impact on me as a student. In particular there are three educators that have contributed a great deal to my education. Firstly, in the last two years both Dr. Salimur Choudhury and Dr. Dr. David Savage have been a constant source of council during the work on this thesis as well as in other projects and studies. The third is Michael Lajoie, who played a large part in sparking my interest in computer science in my undergraduate degree.

Finally I would also like to dedicate this thesis to my fellow graduate students. Whom have provided me with companionship throughout the course of this degree.

Chapter 1

Introduction

A trip to the emergency department (ED) is an event in most people's lives that is universally relatable. At some point, most people have experienced this either first hand, as a patient or by accompanying a friend or family member. Visits to the ED can be highly stressful and the conditions requiring medical attention time sensitive. Providing the right care, to the right patient in a timely manner is essentially a service optimization problem and can have a much greater impact than those implemented in private commercial industries.

1.1 Impacts of Emergency Department Optimization

Optimization of EDs is both beneficial to patient care and at the administrative level. The benefits in quality of a patients stay are very positive, and take the form of increased patient safety and decreased service times. Increased patient safety further results in both, a decreased mortality rate and a decreased revisit rate. Better ser-

vice times will result in shorter physician initial assessment (PIA) and length of stay (LOS), both of these leading to a lower left without being seen rate (LWBS). These factors also have an impact on the patients view on the quality of service forming a more positive community perspective of the hospital. From an administrative perspective better service times, specifically in Ontario where this study takes place, can drastically effect a hospital's funding. In Ontario, all EDs receive base funding dependent on historical patient volumes, additional funding is available to all EDs with good performance metrics, this program is known as the Pay for Results program. This means that EDs that maintain low PIA and LOS times receive additional funding. This funding can then be applied to obtaining further resources, both medical staff and equipment, to even further improve system performance.

1.2 Setting of the Study

Many optimization topics are purposefully designed to be general to ensure a wide range of applications, the unfortunate reality of ED optimization modeling is that each ED is a highly specific instance. This is due to several factors. One of the primary factors being patient demographics. On a macro scale most ED patient demographics will be relatively similar, there are important differences. For example, the shear volume of patients can be different depending on a hospitals location and function (e.g., some EDs will manage trauma and other significant health conditions while others will not). It is due to these variations between departments that the need for individual case studies of EDs rather than constructing a standard method arises. The differing demographics place importance on different resources during a simulation. For example, high trauma EDs will need a larger concern on the part that imaging resources play in the system. In addition there are the administrative choices in how physicians service patients, get assigned to new patients and the handling of triage that can greatly differ between EDs.

This study takes place in the ED at the Thunder Bay Regional Health Sciences Centre (TBRHSC) in Thunder Bay, Ontario, Canada and using the same data that was provided for the study [49]. This a high volume ED and the only trauma centre in northwestern Ontario. There have already been changes to help optimize the ED in the past. The ED is currently divided into two parts; a fast track queue for low acuity patients (e.g., fractured bones, lacerations, and upper respiratory tract infections) and a queue designated for higher acuity patients (e.g., heart attacks, stroke and trauma).

This study is focused on the optimization of the later. In addition, the physician workflow is such that the physicians accumulate patients during the beginning of their shift and then service (i.e., await investigations and provide treatment) these same patients for the remainder. This is different than many other EDs where the physician is continually accumulating new patients throughout the shift. The primary benefit of seeing the majority of patients at the beginning of the shift is that it reduces the likelihood of handover [62]. Handover in this instance refers to the change in the physician responsible for the patients well being during the course of their stay in the ED. Further discussion of these can be found in the related work section.

The optimization problem that is being investigated within this study is physician scheduling. This unlike other approaches that may be considered; such as the benefits of purchasing additional equipment or increasing staffing levels, optimizing scheduling of physicians does not have additional cost associated with it. Therefore the objective is to properly tailor the schedule so that the physicians begin their shifts at times that allow demand to be met the fastest.

1.3 Interacting Components of the Department

During the course of their shifts physicians are a very sought after resource for patients and have little downtime with the addition of other duties [9]. However, this is not the only resource that patients require, creating bottlenecks between components of the department resulting in longer wait times. During the course of a patients visit to the ED the patient does not only interact with the physicians but also auxiliary departments. These interactions are generally due to the need for laboratory testing and imaging procedures. These departments are also responsible for investigations both in the hospital and for some outpatients. In the case of imaging procedures, outpatients have priority over many of those in the ED. This means that the booking of times for inpatients and outpatients can effect the PIA and LOS metrics for the ED as wait times for certain imaging procedures can be rather long. In addition, these metrics can also be affected by other patients visits to the ED. Once all investigations and initial treatments are completed, the physician will make a decision as to whether the patient will be admitted to the hospital or discharged home. In the case of admission the patient must remain in the ED until space can be allocated to them within the hospital, during which time they are occupying valuable resources that other patients may have otherwise utilized. This is referred to as bed blocking. All of

these can greatly impact a patients LOS and potentially the PIA of other patients as a result. Since these are separate departments the ED has to look to optimize around them, as opposed to optimizing them directly.

1.4 Terminology

Due to the fact that this is computer science thesis; and therefore many readers may be unfamiliar with many terms that are used by the medical community mentioned. A full list of descriptions are included below.

- **Emergency Department:** The area of the hospital where a patient is treated when entering with an urgent need and no appointment.
- **Physician Initial Assessment:** The point in a patients stay when they are first assessed by a physician.
- **Length of Stay:** The time from a patients arrival to either their discharge or admission.
- **Left Without Being Seen Rate:** The number of patients who are triaged but leave before their initial assessment by a physician.
- **Mortality Rate:** The number of patients who die during the course of their time in the ED over a particular period of time.
- **Revisiting Rate:** How often patients must return to the ED due to the same complaint over a particular period of time.
- **Acuity:** The severity of a patients condition.
- **Triage:** The process of determining the severity of the patients condition and how quickly they need to be seen by a physician. Patients are given a score from 1 to 5 with level 1 patients requiring immediate attention to prevent harm or death.
- **CTAS Level:** The Canadian Triage Acuity Scale, the system used to triage patients.
- **Patient Handoffs:** When a physician finishes their shift but still has patients that they need to see, instead they are assigned to a new physician.

- **Imaging Procedures:** Procedures that allow for the imaging of a patients internals; CT scans, MRIs, X-rays(radiology), ultrasounds and echo cardiograms.
- **Laboratory Tests:** Tests conducted on blood or urine samples from the patient to determine their condition.
- **Inpatients:** Patients admitted to hospital .
- **Outpatients:** Patients coming from the community to access hospital resources but are not admitted to hospital.
- **Admission:** When a patient is unable to go home for a variety reasons and needs to stay in hospital for further monitoring or treatment.
- **Discharge:** When the patient is deemed fit to leave by the physician.
- **Bed Blocking:** The time a patient spends in the ED after they have been admitted but not yet transferred out of the ED and thus prevent another patient from utilizing that space.

1.5 Thesis Overview

In this remaining section of the chapter a brief summary of the further contents of the thesis will be provided for the reader. In the next chapter, related work, a number of studies will be discussed that helped guide the research of the thesis. These pertain to various methods of optimizing the ED as well as the applications of clustering in the ED. The discussion begins with how academics evaluate ED physician schedules and then proceeds into the methods commonly used to optimize schedules. Following this some changes to EDs on a higher level, that have been shown to be effective in improving metrics, are discussed. Finally areas in the ED that could benefit from the use of clustering to aid physicians in making decisions are provided for the reader.

The two chapters following the related work pertain to the construction of the simulation model. The third, and first of these two chapters, illustrates the process in which patients are generated for use in the simulation model the using provided data. Additionally this chapter provides validation information to argue the quality of these generated patients. The fourth chapter continues into the construction of the simulation model of the ED. This explains the iterations in it's construction and as in the previous chapter provides validation information.

The fifth chapter discusses the cluster partitioning problem. The problem is outlined for the reader and an integer linear programming model is constructed for comparison to the algorithm's performance. Additionally both sequential and parallel versions of the algorithm are provided for the reader. The parallel versions being both CPU and GPU based.

In the remainder of the thesis the results are discussed. In the sixth chapter the results of the simulation and the clustering algorithm's performance are offered to the reader along with the author's comments. In the final chapter some closing thoughts are provided, discussing the results and potential future work.

Chapter 2

Related Work

2.1 Overview

The problem of physician staffing for an ED is a complicated one that is not easily generalized. Schedules must be formed case by case for individual EDs. While there may be similar patterns among hospital ED patient volumes, each EDs structure is decided upon by the hospital governing it. For example, the presence of a fast-track queue for low acuity patients. Another example would be the physician workflow and how physicians accumulate patients over the course of their shift. In some EDs physicians assess new patients throughout their shift and handover those that require further attention at the end. While in other departments, the physicians will acquire patients for an allotted period during the beginning of there shifts and service these patients for the remainder, in order to avoid the handover of patients between physicians. These two simple examples illustrate the very different situations in which EDs operate.

An observational study of physicians was conducted in EDs across Ontatio to measure their activities during shifts [9]. The study consisted of data gathered from eleven hospitals and five different geographic regions. Three different types of EDs were studied: 2 rural, 6 community and 3 teaching hospitals. Data was collected over the course of multiple periods during the course of the year. The results describe the type of patients that visited the eleven EDs, how much time physicians spent doing different activities during the course of their shift and how long physicians spent with types of patients corresponding to Canadian Triage Acuity Scale (CTAS) level. The study found that physicians in EDs in community and teaching hospitals have minimal downtime during the course of their shifts and that CTAS levels present within the EDs varied in their distributions.

2.2 Evaluation of Emergency Physician Schedules

If the objective of a study is to optimize the scheduling of physicians in the ED, then a method for the evaluation of such schedules must also be constructed. Not surprisingly the methods of evaluating ED schedules out date optimization techniques for the area. Evaluations are generally simulation based through the use of discrete event simulation [13], [55], [42] and [45]. A general basis for a simulation tool was developed by [53].

A study was performed at an inner city urban teaching hospital in Vancouver,

British Columbia, Canada, to determine predictors of physician workload [25]. Several key variables were found that predicted total physician time per patient visit, including CTAS, age, sex and whether a medical procedure was required or not. The authors conclude although their model was validated at the same hospital, further validation is required.

In addition to evaluating the performance of current physician schedules against potential schedules, the simulation model can also be employed to test other changes in the ED. Concentration on the minimization of average patient length of stay was found to have adverse affects on other aspects of the department [50]. It was found within the simulation that this minimization resulted in high variability of staff utilization and the length of stay in patients in general. The authors also mention that the common tactic of increasing beds can result in resource bottlenecks. Discrete event simulation was used to investigate the effect of the implementation of a fast track queue and acuity ratio triage assignment in comparison to traditional ED patient assignment [5].

2.3 Optimization of Physician Schedules

2.3.1 Discrete Event Simulation

One method of optimizing physician schedules is through the use of discrete event simulation [15]. Prior to the optimization of the ED's schedule the simulation model is first validated to determine if its an accurate representation of the ED. Typically the optimization of these schedules consists of experts running the simulation for potential schedules. The schedules are then iterively adjusted to produce one that better meets the hospitals needs.

A study was done using discrete event simulation and what if analysis in regards to a hospital in Moncton, Canada [29]. The main objective was the reduction of wait times. The data was collected between the hours of 0800 and 2000 on weekdays only. Additionally, only patients with CTAS levels from 3-5 were considered, as they makeup 93% of patients within the department and levels 1 and 2 were meeting standards. Alternative schedules were explored with the use of additional staff, both nurses and physicians, as well as the use of additional rooms. These scenarios were all constructed with the intent of reducing the time between a patients registration and the availability of an examination room, as it was found to be the most significant

contributor to wait times. The results of the study found that an increase in rooms without a matching increase in staff has no affect on the waiting times. They also show that the addition of a physician and a nurse between 0800 hours and 1600 hours is the most beneficial of the scenarios explored.

2.3.2 Mixed Integer Programming

Another technique is mathematical modeling through the use of mixed integer programming. In this process constraints are created that describe the nature of the ED, such as how many beds and physicians are available. As well, as logical constraints such as only one patient can use a bed at a time. The mathematical solver is then given an objective function, that in this case is generally the minimization of a factor related to ED overcrowding such as: length of stay, waiting time, or patients that left without being seen to name a few.

Stochastic optimization was used by another study in Lille, France [12]. The schedules were evaluated using discrete event simulation and the optimization of schedules was done via a stochastic mixed integer model solved using sample average approximation. Both the simulation and optimization models used exponential service times and the patients arrival was based on a Poisson distribution. However the optimization model is less complex, an example being resource usage including laboratory tests and imaging procedures. Multiple schedules were created using this technique with varying constraints on shift lengths; fixed eight hour shifts, shifts of four to twelve hours and shifts with no length constraint only total hours worked. Optimized schedules were then tested for robustness, simulating a large increase in patient volumes, such as an epidemic. The final evaluation of schedules was done by simulating 100, ten day periods for each of the the three generated schedules as well as the original. The authors found that the additional flexibility of no required shift length was not more beneficial than that of a four to twelve hour shift, and that the most important factor overall in the optimization of a schedule is the start time.

A study was done using historical data from a teaching hospital in Thunder Bay (Thunder Bay Regional Health Sciences Centre), Ontario Canada that used mixed integer programming to optimize a schedule [49]. Temporal patterns within the data where analyzed between days of the week in regards to patient arrival rates. The authors found that a division of a schedule into one for weekdays and one for weekends was best. The model was constructed illustrating constraints in the ED including the

two queues (i.e., a fast track and acute care), and accounts for movement of physicians from the acute care queue to the fast track during the last three hours of their shift. During the course of the study three scenarios were explored; a schedule that was simply an optimized version of the current one, a schedule that made use of an additional physician in the acute care area of the department and a schedule that used an additional physician in the fast track area. The model was able to generate better performing schedules for all three scenarios. The scenario of a revised schedule with no additional physicians reduced the unmet patient demand by 19%. Note, the unmet demand was calculated as the average number of arriving patients beyond the physician productivity. As one may expect it was found that an additional physician reduced unmet demand further. Specifically having the greatest effect in the the fast track area. However the authors mention that the choice to utilize the physician in the acute care area may be better when other factors are considered.

2.3.3 Algorithmic

A more recent area that is being explored is optimization through the use of algorithms [51]. These algorithms tend to borrow the generalized framework of resource scheduling algorithms and modify them to fit the parameters of the ED.

A study was done that constructed two iterative algorithms that made use of a linear optimization model [52]. The study was conducted using data gathered from five EDs in Israel, of which one was a level 1 trauma centre, two were medium sized hospitals and two were small hospitals. The physicians in the study were divided into categories within the ED. The two algorithms proposed were largely similar with the second being an extension of the first. The algorithms attempt to reduce the average patient length of stay by finding the largest contributor within the department and rescheduling that area accordingly. The schedule is tested between iterations using a simulation model. The first algorithm ignores any areas that are scheduled that do not have more than 24 person hours assigned to them as they are are unable to be rescheduled. While the second algorithm allows these areas to be rescheduled when staff from other areas can be borrowed, choosing the one that is causing the least delay. The results of the algorithms showed that across the data from the five hospitals there was an average reduction in the length of stay of patients between 7% and 17.5% for the first algorithm, the second showing between 11% and 29%. The authors offer some areas that could be potentially explored for further benefit.

Firstly, the modifying of the algorithm to incorporate mixed shift lengths, as currently it only considers eight hour shifts. Additionally, changing the heuristic so that it also considers variability of wait times, as it could yield a more robust schedule. Finally the consideration of cost functions, as shifts can have different cost depending on time.

2.3.4 Queuing Theory

Because an ED is essentially a service system, queueing theory has been offered as a possible means of optimization [18]. A study was done that constructed a queueing model of an urban ED in Manhattan [17]. The model considered a single queue for entering patients that fed into servers (i.e., the staff) with constant arrival rates and service times using an exponential distribution. To account for the fact that EDs having varying arrival rates through out the day, patient flow was broken down according to a lag stationary independent period by period (SIPP) approach. This is modified from the standard SIPP to account for the fact that peak congestion often occurs slightly after peak arrival in many service systems. Each of these periods is then solved independently to find the required minimum staffing level to meet targets. The objective in question was based on the amount of patients that left without being seen, and is that a patient will not have a probability higher than 20% of waiting an hour to be seen. To investigate the effects of the optimization, data was compared from a 39 week period before the schedules implementation and a 39 week one after the implementation. These periods consisted of matching weeks to account for seasonal and disease state variation. Also being aligned to account for days of the week. Schedule construction was divided into weekdays and weekends. Between the two periods an increase in patients of 6.3% occurred, while the new schedule resulted in a left without being seen percentage of 6.4% opposed to the previous 8.3%. The authors also comment that for the periods of the week where the number of staff hours was unchanged, just the scheduling had a left without being seen percentage of 7.2% compared to the previous 9.2%, despite a 5.5% increase in patients during the time. The authors note that these results could be further improved if proper data was provided on the length of time providers spent with patients and patient triage level, as well as the effects of patients waiting on test results.

2.4 Advantages of Physicians in Triage

Studies have investigated the effects of modifying the traditional triage process by having a physician during this early stage of a patients stay in an ED [35]. The idea behind this being that physicians can more reliably determine a patients acuity and then order the required investigations while the patient waits to access a treatment space. In addition this approach is usually considered when the objective of the optimization is focused in the earlier portion of a patient's visit.

Having a physician involved in the triage process has been shown to decrease the time spent waiting to see a physician, left without being seen rate and length of stay [31], [24], [39], [16], [22] and [10]. The presence of a physician in triage allows for low acuity patients not needing laboratory tests or imaging to be discharged immediately after the triage stage [58]. In addition to reducing waiting times for all patients, having a physician in triage allows physically small EDs to service patients while no beds are available [40]. With the possibility of having laboratory tests and imaging ordered at triage, physicians in triage allow for less time spent waiting in a bed in the ED [47] and [56]. This is due to the fact that without a physician in triage a patient would have to first wait for an initial assessment by a physician before having the tests and imaging ordered.

A study was done at an urban academic medical centre in San Diego, California, in which the effects of their implemented REACT (rapid entry and accelerated care at triage) system had on the ED were analysed [2]. REACT created many changes for the process of ambulatory patients, which make up 85% of their patient population. Among the changes was the allowing of patients to have a medical record started prior to full registration. Allowing this opened the door to several possibilities. This allows for such things as tests being ordered and the patient being immediately sent to available rooms after triage. In addition if no beds in the ED were available, nursing staff were directed to contact the physician, who could then make a brief evaluation at triage and order lab or radiology tests. This change in the system allows for tests to be completed while the patient is waiting for a full registration, where as previously none of this would have even begun. The implementation of REACT resulted in a decrease of the left without being seen rate of 50%, those that received accelerated testing and care at triage made up 8% of the total population and that 23% of patients waited five minutes or less to be seen.

Another study investigated the effects of having physician at triage in a urban

academic level 1 trauma centre in a medium sized city [20]. Data was collected during a 9 week period before and after the implementation, forming a control data set as well as one effected by the presence of a physician at triage. With the control data collwction ending one day prior to the data during the test period. This was done in order to minimize temporal and seasonal changes. The LWBS rates decreased from 4.5% to 2.5%, with no change in patient volumes. Ambulance diversion also decreased, from 5.6 days per month (36 episodes) to 3.2 days per month (29 episodes), with a decrease in duration as well from a median of 431.5 minutes per episode to a median of 256 minutes per episode. Decreases in patients length of stay however were only found with discharged patients, due to the boarding times within the department.

2.5 Effect of Fast Track Areas

As previously mentioned in section 2.1 some EDs make use of fast track areas. The function of these areas are to specifically target the length of stay for lower acuity patients. While at first glance this may appear in opposition to the triaging system, as patients are prioritized based on severity of acuity. This however is untrue as it allows low acuity patients to continue flowing through the ED while beds are being blocked by patients that have been admitted but not yet left the ED. In addition to this it has also been found to reduce the left without being seen rate and length of stays in several EDs [43], [23], [38] and [14]. These fast track queues are not always managed by physicians, sometimes by nurse practitioners or a combination of physicians and nurse practitioners. This can be beneficial from a budgetary perspective, since having low acuity patients being handled by a nurse practitioner is less expensive than a physician. In addition to this budgetary advantage, by having lower acuity patients be serviced, at least in part, by nurse practitioners it allows physicians to concentrate on the higher acuity patients [26]. Studies have been done that show that patients are very happy with the service provided by nurse practitioners within fast track areas [8] and [30]. It has also been found that fast track patients have less tests that need to performed, meaning that the fast track area's patient flow is less effected by other departments [19]. It has been shown that these effects still persist with increased amounts of patients and therefore an increased workload for physicians [27].

The introduction of fast track areas in EDs has been shown to decrease the number of patients with lengths of stays of 4 or more hours [4]. A fast track area was shown to reduce the number of patients waiting over an hour by 30%, and 50% with an

increased consultant presence [7]. These decreased length of stays lead to an increase of patient flow and decreased likelihood of overcrowding [28]. As mentioned, a concern that is raised when the implementation of fast track areas is considered is the effect it will have on higher acuity patients. It has been shown that in addition to lowering waiting times for lower acuity patients, there are also benefits for those with higher acuity [44].

One study done at an urban tertiary ED that services 75,000 patients per year investigated the effectiveness of the newly implemented fast track area [48]. The study was done by analysing data from a period prior to the implementation of the fast track area and a period after. The second period experienced a 4.43% increase in daily patients compared to the first and 30% of the total patients were triaged to the fast track area. There were clear benefits to the ED from implementating the fast track area. There was an average decrease of 50% in the waiting time too see a physician, including both fast track and non-fast track patients. There was a decrease in the average length of stay by 9.79% on average, again both fast track and non-fast track. The authors mention that the primary concern is that the implementation of the fast track area would effect the quality of care. To address this the authors point out that there was a decrease of 52.18%, 1.31% and 3.57% in the left without being seen rate, revisiting rate and mortality rate, respectively.

A similar study was done at a tertiary adult ED in Perth, Western Australia, with a trial period for a fast track department [36]. This ED experiences a large number of elderly patients (i.e., 26%) over 70 years of age and 14% over 80 and of the total patients 48% were admitted. The trial period consisted of a twelve week period of an operational fast track area between 09:00 and 22:00 on weekdays and 09:30 and 18:00 on weekends. This area was staffed by a single junior physician and a single nurse, with no increase in staffing levels within the department. Triaged patients were assigned to the fast track area when they where expected to be discharged and had low acuity scores, 3 to 5 on the Australian Triage Scale. Over this period the fast track area handled 21.6% of patients, 123.5 per week on average, and 29.8% of all patients who were discharged. This trial period resulted in the reduction of both length of stay and waiting times for discharged patients. The relative decrease in length of stay was shown to be 18% and 9.7%, for the matching 12 week period of the previous year and the 12 week period preceeding the trial, respectively. These decreases were in the face of 7.7% increase in patients from the previous year and a 10.2% seasonal increase in patients. Similarly there was a relative decrease in the waiting time to

see a physician for the prior periods of 20.3% and 3.4%, respectively for discharged patients. There was also a relative decrease in patients who left without being seen of 37% and 17% for respective periods. The authors mention that there were no increases in the average waiting time for admitted patients with either period. The fast track area used during the course of the trial was allocated 3 beds in comparison to 500 in the standard ED. The authors mention that there are limitations in the study. During the course of the of the matching period of the previous year the ED was expanded, providing additional physical space, hence the investigation of the two prior periods to try to negate any reductions that were a result of increased area.

A study was done at a teaching hospital in Melbourne, Australia to investigate the effectiveness of a fast track area within the ED [6]. Data was gathered over the course of two periods, July 1st 2006 to November 15th and January 1st 2007 to March 31st 2007, being a ED setting without a fast track area and one with a fast track area respectively. The implementation of the fast track area resulted in a decrease in length of stay for discharged patients and no significant change with admitted patients. Both the periods prior and following the implementation of the fast track area had 14% of discharged patients have a length of stay of 60 minutes or less. This diverges in the cases of patients who stay 2 hours or less. This accounts for 44% of non-fast track discharged patients and 53% of fast track patients. Similarly in the case of patients who stay 4 hours or less. Accounting for 84% of non-fast track patients who where discharged and 92% of the fast track patients. The authors mention that the demographic and volume of patients did not significantly change within the two periods.

2.6 Managing Patient Handovers

Patient handovers are another area of research that can be used to improve EDs. Patient handovers occur when a physician has finished their shift but patients under their supervision require further attention before being admitted or discharged. A study was done in which the handovers of 992 patients were observed at an urban teaching hospital over an 8 week period [33]. During this period it was found that physical examination errors occurred in 13.1% of cases and omissions in 45.1% of cases. While laboratory errors and omissions were found in 3.7% and 29.2% of cases respectively. A similar study was done with the more specific focus on vital sign communication at an urban academic tertiary care hospital [59]. The study observed

1163 patient handovers, in which 42% of those with episodes of low blood pressure (66 of 117) and 74% of those with an episode of low levels of oxygen in the blood (116 of 156) were not communicated to the physician taking over patient care. The authors mention that even a single episode of low blood pressure is associated with higher mortality rates. The study also found that omissions of vital sign occurred in 14% of handovers. Further support to the errors present in handovers can be found in [61]. These errors often lead to increased patient stay that can be a result of the new physician performing examinations that were performed by the original.

A general framework for patient handovers is discussed and presented in [3]. The authors discuss potential causes of handover errors between physicians. They mention that the ED is a chaotic work environment that can lead to interruptions during the handover process resulting in errors. In addition to this since time is a valuable resource in the ED errors can occur from physicians balancing conciseness and completeness when performing handovers. Other sources mentioned are that many physicians will continue to perform activities such as charting leading to confusion about which physician is responsible for specific tasks, as well as poor communication of factors such as pending results of imaging, laboratory tests and consultants and unclear diagnosis. The author's offer general guidelines to better streamline the process of handovers. These include; a dedicated space for the process of handovers to reduce distractions, a structured overview of patients with initial assessment, imaging and laboratory results including those that are outstanding, properly accounting for patients temporarily in other departments, the establishment of clear moments of transfer.

As mentioned in the beginning of this chapter some EDs organize their schedules in a manner that a physician shift has two phases; the first where they acquire new patients and the second where they mainly service their existing patients. Organizing shifts in such a manner reduces the number of handoffs necessary during a physicians shift. A study was conduct on the effects of such a scheduling change at a the Seattle Children's Hospital [62]. The study observed 43,835 patient encounters, in which patient handoffs where reduced from 7.9% to 5.9%. Surveys also showed improved perceptions of patient safety, patient flow and job satisfaction.

2.7 Potential Benefits of Clustering

We know arrive at the secondary area of investigation within the study, clustering. In this section some potential benefits of clustering will be discussed in regards to the ED.

2.7.1 Patient Predictions at Triage

Clustering can yield some advantages if utilized during the triage process. It can be used as both a method of determining the seriousness of a patients condition and to get a prediction of resources that a patient may require throughout their stay. The determining of the seriousness of a patient's condition with the aid of clustering can help to more effectively assign patients that are borderline to queues when a fast track queue is present within the ED. When used to determine resources that patients are likely to require throughout their stay in the ED, wait times for things such as laboratory tests and imaging procedures can be drastically cut down as the process can begin while the patient is waiting for their initial assessment. One study developed a clustering system to show a patients association with resources and admission with regards to presenting complaints using hierarchical clustering [34]. This allowed patients to be categorized into three groups of acuity. Another study used data from a hospital in west London in the UK to determine between to groups of high acuity patients [32]. This study used both K means clustering and fuzzy C means. A study was done using hierarchical clustering that identified misdiagnoses of influenza case for respiratory disease [54]. The authors stress the reduction in overcrowding that could be seen if this was used during flu seasons.

2.7.2 Building Clinical Profiles for Patients

While all uses of clustering involve building profiles for patients to a degree there are certain situations where additional information can be utilized by physicians through the use of patient profiles alone. These are often highly specified but if introduced as a system within the ED they could be very useful to physicians. One study used data from the Massachusetts General hospital to make accurate predictions for septic shock in patients [37]. This was done using agglomerative hierarchical clustering of blood pressure trajectories. Another study was done using K means clustering to categorize patients seriously attempting suicide at a University hospital in Brazil [41].

The study was able to determine three groups of patients that are likely to require admission, although the authors mention that further analysis is needed.

2.7.3 Predicting Likelihood of Admission

Another application of clustering is determining whether a patient will require admission. If patients that are to be admitted can be identified earlier then fewer patients will be occupying treatment spaces within the ED. If time to admission can be reduced it could potentially aid in overcrowding problems within the ED. One study used ward's method to cluster patients in a psychiatric ward to predict admission [1]. Another study focused specifically on patients having CT head scans and found sub populations that were likely to be admitted [57]. This paper used data from three Emory hospitals in Atlanta to perform K means clustering.

2.7.4 Length of Stay Related Benefits

Clustering can also be applied in regards to a patients LOS. Some studies focus on predicting a patients LOS. Such as one study that uses K means clustering on the MIMIC II data set [46]. It was able to predict death and LOS of patients. Another use of length of stay is not through predicting it but rather using previous data to estimate future resource consumption. One study used a variety of clustering techniques on a stroke victim data set from the English Hospital Episode Database to do just this [11].

2.7.5 Frequent User Profile Identification

A less obvious use of clustering is the construction of frequent user profiles. Some patients have conditions that often require medical attention. These can be consistent visits or short bursts periodically. Being able to build these profiles can allow an ED to perform forecasting to a better manage these populations. Additionally it could allow the ED to reevaluate their approach with these patients as at times admission could remove the need for additional visits for a period. One study used decision trees to identify patients likely to return within 30 days [21]. This resulted in the identification of high risk patients and patterns in resource consumption. Another study was done using spectral clustering with Wasserstein distance to identify and build profiles of frequent users that take a toll on the ED [60].

2.8 Chapter Summary

In this chapter various methods that have been used to optimize EDs have been discussed. There are both approaches that focus on merely improving how schedules meet patient demand and those that change the way the ED functions as a whole.

In regards to scheduling physicians the current common practice for evaluating schedules is through the use of discrete event simulation. This is done in order to judge whether or not a schedule properly meets patient demand. In order to actually optimize the physician schedules, surveying the research in this area showed four different approaches. The first approach was to use discrete event simulation to evaluate a handful of schedules. This is often done to determine if small changes should be made to schedules or to choose between different proposed schedules. Alternatively the schedules are adapted through what if analysis in order to determine a more efficient schedule. The second approach that is commonly used is mixed integer programming. In this approach a model is defined through the use of equations that constrain the problem space. Algorithmic approaches are also used. These consist of using simulation to evaluate schedules and a heuristic to guide the optimization. Finally the fourth common approach is the use of queuing theory. This approach models the ED in the form of a service problem.

Following this, approaches that introduce changes to the way the ED functions were discussed. The first approach was the use of physicians in triage. This provides more accurate triaging of patients and allows for the ordering of laboratory tests and imaging procedures immediately. The second approach discussed was the effects of implementing a fast track area within the ED. These areas allow low acuity patients to be seen quickly instead of waiting for high acuity patients to free up necessary staff and resources. Introduction of these areas has been found to have beneficial effects on key performance metrics for EDs, namely PIA and LOS. Last managing the handover of patients at the end of a physicians shift was discussed. Common issues that result from the miscommunications between physicians where discussed as well. Additionally the approach to remedy these issues of having physicians work in two phases was offered to the reader. This approach first has the physician focus on acquiring new patients for the first portion of their shift and then attending to these patients for the remainder of the shift.

Finally the potential benefits of utilizing clustering in the ED are discussed. The studies show the ability of clustering to produce accurate predictions for patients

from information obtained at triage and the capability to build clinical profiles. Additionally clustering can be used to determine if patients are likely to need admission, which could greatly decrease bed blocking times, and the estimated LOS of patients. Lastly clustering can be used to identify likely frequent users of the ED.

Chapter 3

Modeling the Patients

The first step in modeling the ED is properly generating patients for a simulation. These generated patients need to be representative of those that visit the ED. Therefore they can not be simply randomly generated and need to be based of the data.

3.1 Provided Data

The data provided by the TBRHSC comes from two sources; the ED and radiology. The first dataset contains all of the patient descriptors and several time stamps that denote important processes within the ED (e.g., arrival, physician initial assessment, investigation ordering, admission and/or discharge). The second data set only contains information for laboratory tests and imaging procedures.

The information in the ED data that are made use of are as follows:

- **Time and date of the patients arrival**

- **Time and date of the patients PIA (point of initial assessment)**
- **Time and date of the patients end of stay(this may be discharge or admission)**
- **Time and date of the patients transfer to the main hospital(this is in the event of admission)**
- **The patients primary complaint upon arrival**
- **The patients age**
- **The patients sex**
- **The patients assigned Canadian Triage Acuity Scale(CTAS) level, rankings of this scale can be seen in figure 3.1**
- **The patients area of treatment within the ED**

CTAS 1	Resuscitation
CTAS 2	Emergent
CTAS 3	Urgent
CTAS 4	Less Urgent
CTAS 5	Non Urgent

Figure 3.1: CTAS levels.

The information in the radiology data that are made use of are as follows:

- **The name of the laboratory test or imaging procedure**
- **The category, in the case of imaging procedures**
- **The time and date that the order was placed**
- **The time and date of collection, in the case of laboratory test**

- **The time and date when the results became available, in the case of laboratory tests**
- **The time and date where the patient enters the imaging procedure**

A unique patient ID was also provided to ensure patient anonymity and to link the two datasets.

3.2 Process of Generating Patients

For the purpose of modeling the ED synthetic data is used. These process of generating these patients is elaborated on in the rest of this chapter. The goal is was to have the demographics and resource needs of these generated patients to match the records found in the data as close as possible. The reason for doing this is two fold. First it avoids the issue of partially incomplete records for patients. These patients would need to be removed from the simulation making the ED less busy then it truly was. Second it allows for the generation of multiple sets of data to aid in tuning the simulation parameters.

While several of the variables can be clearly used in the process of generating patients, the use of several other variables was not. In particular, the inclusion of many of the date and time data points. These dates and times were included as a method of double checking those used in the generation process. As this data is hand entered by hospital staff throughout the day there are some obvious entry errors in the data. The errors can be as simple as transposition errors that mix up the date format so that the so that the month and day are changed. In the instances where this creates an impossible date, for example the 2nd day of the 20th month, this is an simple mistake to catch. However, if the date is still possible then the date and time stamps can be checked against other events in the patients stay to ensure that a logical order is followed. This process can also be used to detect errors in the time stamp as well.

3.2.1 Time of Arrival

The generated patient arrival times must accurately match patient times and volumes in the data. Another consideration is whether the data should be split to account for differences in patient arrivals between weekdays, weekends, and holiday days seen

in previous research [49]. Several holiday dates were removed from the data set to reduce variability. These were long weekends in which the Friday or Monday where removed along with the dates that directly proceed and follow them. This was done with the thought that these days may behave closer to the weekend days and may skew the result.

The patient arrivals were then divided into 96, 15 minute increments throughout the day. These begin on the hour, the quarter hour, the half hour and the three quarter hour. The mean number of patients arriving in the ED was then calculated, which are then used in a non-stationary Poisson process. This generated results very close to the true means but some further adjustments where necessary and arrivals where dropped randomly form the farthest outlier of each group, until the correct mean was reached. The algorithm for the Non-Stationary Poisson Process can be seen bellow in algorithm 1. This process was done for both weekdays and weekends separately.

Algorithm 1 Partitioning Algorithm

```
for number of minutes to generate patients for do
    a: number of patients who arrived in the last 15 minutes
    for j in the past 15 minutes do
        if j ¡ 0 then
            continue  this is a boundary condition
        end if
        if a patient arrived during minute j then
            a+=1
        end if
    end for
    if a >= 6 then
        continue  realistic bound from looking at data
    end if
    r1 = probability of at least a+1 patients having arrived in the past 15 minutes

    r1 = probability of at least a patients having arrived in the past 15 minutes
    chance = r1/r2
    generate a random number to see if there is and arrival
    if there was an arrival then
        record the arrival time
    end if
end for
```

3.2.2 Patient Profile

With the patient's arrival time generated, the patient's characteristics must be generated. The variables required for each patient are: chief complaint, sex, age, CTAS level, laboratory tests, imaging procedures and admission/discharge. During the course of the simulation the type of laboratory test or imaging procedure required is irrelevant, it is whether it none, one, or both occur. Consecutive laboratory tests were not uncommon in the actual ED and may be needed for monitoring reasons. These can not be easily generated as their are some causal relationships at play.

The first set of important relationships is that of age, sex and chief complaint. While all sex and age combinations are valid, extreme old ages are less likely. Chief

complaints are not valid with all combinations of age or sex. For example, a fever can be serious for a baby but may not be significant for an adult, indicating that age is a determiner of chief complaint. An example, where sex is a determiner would be cases that relate to female sexual organs, such as uterine bleeding which a biological male cannot have. Frequency distributions for each combination sex, age, and chief complaint were developed, negating ones that were not present. The patient age was binned in five year intervals in order to decrease the resolution of the data and eliminate "noise" within the data. These tuples where then drawn from the distribution according to their relative weights in the data.

At this point our patient has the descriptors age, sex, and chief complaint. The next step was to generate a CTAS level for each patient. The CTAS level of the patient is directly related to the previously generated descriptors as that is what a nurse will primarily use to triage the patient upon first assessing them. We know that these relationships are important but some are likely to be less important than others and can be essentially drawn from the global CTAS distribution. To determine the important relationships we borrow the concept of association rule mining from the field of big data. To accomplish this analysis, the support and confidence for each combination must be calculated. The equation for these to metrics can be found in equations 3.1 and 3.2, respectively. When the combination of minimum support and confidence is used, a relationship in the number of association rules generated emerges, this can be seen in figure 3.2. The chosen minimum support and confidence were 4.0x10^-8 and 4.0x10^-8 respectively. The relationships not deemed important are simply drawn from the global CTAS distribution with the caveat that there must be an example of the combination within the data. For those deemed important, a chance is first given to this relationship by using the confidence.

$$support = \frac{occurrences\ within\ data}{amount\ of\ data\ points} \tag{3.1}$$

$$confidence = \frac{occurrences\ with\ both\ the\ antecedent\ and\ consequent}{occurrences\ with\ the\ antecedent} \tag{3.2}$$

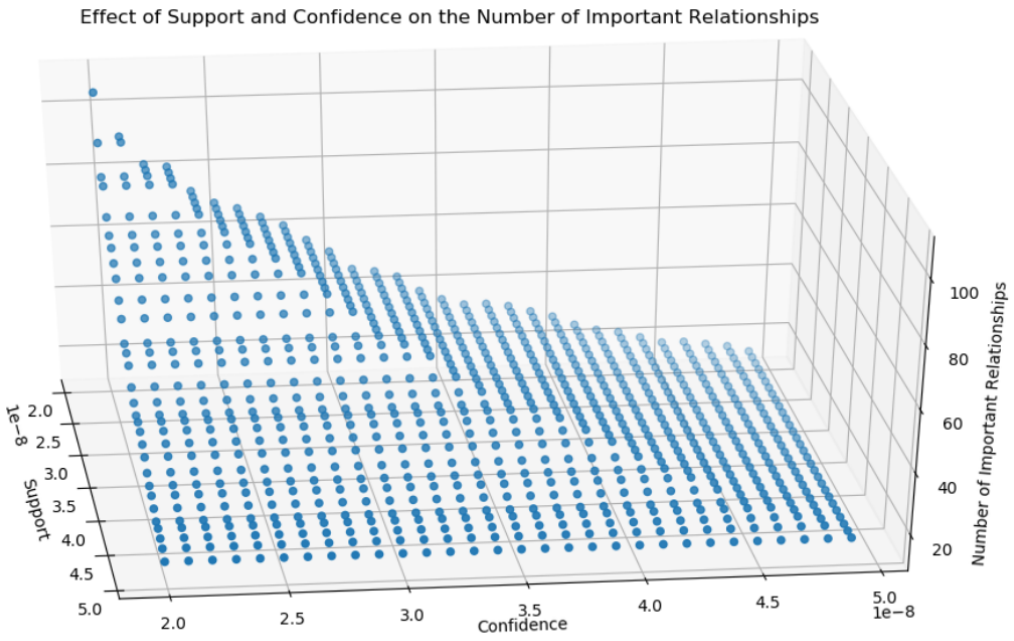

Figure 3.2: Important relationships between (chief complaint, age bin, sex) and CTAS Level.

Next, the orders for laboratory test and imaging procedures were generated. For our purposes we do not need to know what the type of test or procedure, but instead that one is required. To do this we again use the Poisson distribution to determine whether the number of rounds of tests or procedures the patient undergoes. For these distributions we need to determine the probability for each combination of chief complaint, sex, age bin and CTAS level ordering a procedure or test. As well as how likely it is to order subsequent ones. We again use the timestamps within the data to ensure that the test or procedure was completed prior to admission. Note that due to the extremely small number of instances, the MRI and echocardiogram procedures were not considered for the simulation. For each of the patients, it is then determined how many laboratory test, CT, radiology and ultrasound rounds they receive before being admitted or discharged. It should be noted that there are possibilities within the data for multiple tests or procedures to be ordered at once, this is extremely common with lab tests. Taking this into consideration, before determining the number of rounds a patient underwent, they were grouped by the end of the test or procedure as the patient would be waiting until that point so it did not matter if an additional test or procedure of the same category was added to the list between then if it was

in the same batch of completed results.

Finally, the patient is either admitted or discharged. This is done simply by using the proportion of the patients in the data that are admitted or discharged according to their chief complaint, sex, age bin and CTAS level.

3.3 Validation of Generated Patients

Before starting the modeling, the generated patients were compared to the true data. To do this the steps will be compared in a step by step fashion.

To begin we have the patient arrivals generated for simulation. A comparison between the generated patients and those from the data can be seen in figure 3.3. These results where obtained by simulating 365 days of patient arrival.

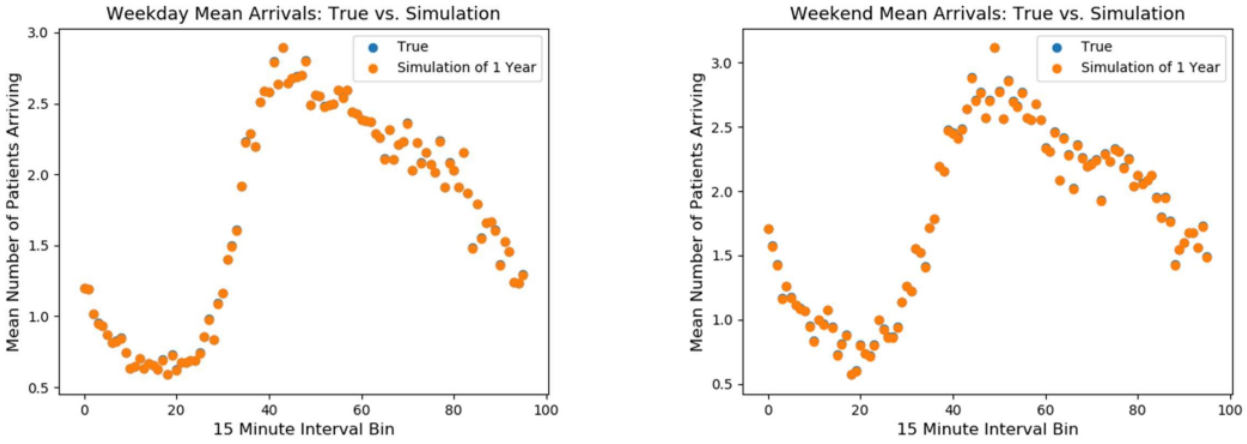

Figure 3.3: Simulated patient arrivals over a 365 day period compared to those in the data.

Following this we have the generated patient descriptors. The comparisons for age bin and sex can be seen in tables 3.1 and 3.2, respectively. The comparisons for chief complaints can be seen in table 3.3. This table only contains the top 20 chief complaints for the sets, as there are 177 chief complaints within the data. However, these top 20 makeup about 63% of the patients in both the data and the generated set.

Table 3.1: Comparison between data and generated binned age proportions.

Age Bin(Years)	Data Proportion	Generated Proportion

0-4	6.64%	6.75%
5-9	1.73%	1.77%
10-14	2.33%	2.29%
15-19	5.28%	5.37%
20-24	7.80%	7.80%
25-29	6.71%	6.72%
30-34	5.99%	5.79%
35-39	5.93%	5.86%
40-44	5.36%	5.37%
45-49	5.72%	5.73%
50-54	6.44%	6.50%
55-59	6.73%	6.59%
60-64	6.11%	6.08%
65-69	6.10%	6.05%
70-74	5.44%	5.54%
75-79	4.71%	4.77%
80-84	4.51%	4.54%
90-94	3.68%	3.67%
95-99	2.13%	2.18%
100-104	0.63%	0.59%
105-110	0.04%	0.03%

Table 3.2: Comparison between data and generated sex proportions.

Sex	Data Proportion	Generated Proportion
Male	44.69%	44.91%
Female	55.31%	55.09%

Table 3.3: Comparison between data and generated chief complaint proportions for the the top 20 most occurring.

Rank	Data Complaint	Data Proportion	Generated Complaint	Generated Proportion

1	ABDOMINAL PAIN$_N$*ON* – *SPECIFIED*	11.73%	ABDOMINAL PAIN$_N$*ON* – *SPECIFIED*	11.36%
2	SHORTNESS OF BREATH/DYSPNEA	5.51%	SHORTNESS OF BREATH/DYSPNEA	5.57%
3	CHEST PAIN CARDIAC	5.37%	CHEST PAIN CARDIAC	5.19%
4	COUGH	4.19%	COUGH	4.23%
5	BACK PAIN	3.61%	BACK PAIN	3.52%
6	NAUSEA AND/OR VOMITING	3.38%	NAUSEA AND/OR VOMITING	3.29%
7	INTOXICATION/SUBSTANCE MISUSE	2.99%	INTOXICATION/SUBSTANCE MISUSE	3.00%
8	URINARY TRACT INFECTION	2.81%	URINARY TRACT INFECTION	2.86%
9	HEADACHE	2.44%	HEADACHE	2.46%
10	FEVER	2.38%	FEVER	2.35%
11	LOWER EXTREMITY PAIN	2.15%	MINOR COMPLAINTS NOT SPECIFIED	2.22%
12	MINOR COMPLAINTS NOT SPECIFIED	2.09%	LOWER EXTREMITY PAIN	2.19%
13	ANXIETY/SITUATIONAL CRISIS	2.07%	ANXIETY/SITUATIONAL CRISIS	2.10%
14	HEAD INJURY	2.02%	HEAD INJURY	2.10%
15	FLANK PAIN	1.92%	FLANK PAIN	2.00%
16	WEAKNESS/FATIGUE	1.91%	WEAKNESS/FATIGUE	1.97%
17	VERTIGO/DIZZINESS	1.70%	VERTIGO/DIZZINESS	1.88%
18	UPPER EXTREMITY INJURY	1.55%	UPPER EXTREMITY INJURY	1.60%
19	CHEST PAIN$_N$*ON* – *CARDIAC*	1.51%	CHEST PAIN$_N$*ON* – *CARDIAC*	1.47%
20	SYNCOPE/FAINT	1.47%	SYNCOPE/FAINT	1.44%

Next the comparisons for CTAS levels can be seen bellow in table 3.4.

Table 3.4: Comparison between data and generated CTAS level proportions.

CTAS Level	Data Proportion	Generated Proportion
1	3.35%	1.85%
2	36.71%	36.60%
3	56.12%	59.48%
4	3.51%	2.03%
5	0.32%	0.04%

Similarities in the ordering of laboratory test, CT scan, radiology scan and ultrasound orders can be seen in tables 3.5, 3.6, 3.7 and 3.8 respectively. While the admission to discharge ratios can be seen in table 3.9.

Table 3.5: Comparison between data and generated laboratory test order rates.

Number of Orders	Overall Data	Overall Generated
1	57.53%	57.80%
2	11.61%	11.40%
3	3.34%	1.98%
4	0.58%	0.49%
5	0.19%	0.14%
Number of Orders	CTAS 1 Data	CTAS 1 Generated
1	74.76%	74.62%
2	32.56%	32.39%
3	12.31%	11.00%
4	5.00%	4.79%
5	2.06%	1.51%
Number of Orders	CTAS 2 Data	CTAS 2 Generated
1	64.67%	63.40%
2	15.65%	14.17%
3	3.24%	2.67%
4	0.76%	0.66%
5	0.25%	0.18%
Number of Orders	CTAS 3 Data	CTAS 3 Generated

1	53.63%	54.39%
2	8.23%	9.26%
3	1.10%	1.32%
4	0.23%	0.26%
5	0.06%	0.07%
Number of Orders	CTAS 4 Data	CTAS 4 Generated
1	31.76%	41.40%
2	3.98%	5.41%
3	0.74%	0.73%
4	0.15%	0.24%
5	0.00%	0.00%
Number of Orders	CTAS 5 Data	CTAS 5 Generated
1	24.04%	52.17%
2	4.92%	0.00%
3	0.00%	0.00%
4	0.00%	0.00%
5	0.00%	0.00%

Table 3.6: Comparison between data and generated CT scan order rates.

Number of Orders	Overall Data	Overall Generated
1	14.76%	14.84%
2	8.06%	8.08%
3	0.40%	0.35%
4	0.03%	0.02%
5	0.00%	0.00%
Number of Orders	CTAS 1 Data	CTAS 1 Generated
1	30.50%	31.14%
2	12.67%	15.88%
3	2.52%	2.48%
4	0.21%	0.18%
5	0.00%	0.00%
Number of Orders	CTAS 2 Data	CTAS 2 Generated

1	19.53%	19.11%
2	10.33%	10.48%
3	0.59%	0.55%
4	0.03%	0.01%
5	0.00%	0.00%
Number of Orders	CTAS 3 Data	CTAS 3 Generated
1	11.46%	12.04%
2	6.68%	6.50%
3	0.18%	0.17%
4	0.01%	0.01%
5	0.00%	0.00%
Number of Orders	CTAS 4 Data	CTAS 4 Generated
1	3.53%	5.25%
2	2.45%	4.20%
3	0.00%	0.00%
4	0.00%	0.00%
5	0.00%	0.00%
Number of Orders	CTAS 5 Data	CTAS 5 Generated
1	3.28%	0.00%
2	0.00%	0.00%
3	0.00%	0.00%
4	0.00%	0.00%
5	0.00%	0.00%

Table 3.7: Comparison between data and generated radiology scan order rates.

Number of Orders	Overall Data	Overall Generated
1	41.21%	41.79%
2	1.79%	1.69%
3	0.17%	0.15%
4	0.04%	0.04%
5	0.01%	0.00%
Number of Orders	CTAS 1 Data	CTAS 1 Generated

1	60.79%	59.54%
2	6.96%	5.77%
3	1.49%	2.22%
4	0.62%	1.24%
5	0.1%	0.18%
Number of Orders	CTAS 2 Data	CTAS 2 Generated
1	50.69%	46.28%
2	2.24%	2.25%
3	0.19%	0.16%
4	0.04%	0.03%
5	0.00%	0.00%
Number of Orders	CTAS 3 Data	CTAS 3 Generated
1	34.96%	38.82%
2	1.27%	1.26%
3	0.08%	0.08%
4	0.01%	0.01%
5	0.00%	0.00%
Number of Orders	CTAS 4 Data	CTAS 4 Generated
1	25.63%	32.36%
2	0.64%	0.48%
3	0.00%	0.00%
4	0.00%	0.00%
5	0.00%	0.00%
Number of Orders	CTAS 5 Data	CTAS 5 Generated
1	14.21%	21.74%
2	0.00%	0.00%
3	0.00%	0.00%
4	0.00%	0.00%
5	0.00%	0.00%

Table 3.8: Comparison between data and generated ultra sound order rates.

Number of Orders	Overall Data	Overall Generated

1	9.80%	9.95%
2	2.14%	2.17%
3	0.11%	0.12%
4	0.00%	0.00%
5	0.00%	0.00%
Number of Orders	CTAS 1 Data	CTAS 1 Generated
1	3.40%	5.06%
2	0.31%	0.62%
3	0.00%	0.00%
4	0.00%	0.00%
5	0.00%	0.00%
Number of Orders	CTAS 2 Data	CTAS 2 Generated
1	8.40%	9.71%
2	1.69%	2.25%
3	0.07%	0.07%
4	0.00%	0.00%
5	0.00%	0.00%
Number of Orders	CTAS 3 Data	CTAS 3 Generated
1	11.37%	10.28%
2	2.60%	2.17%
3	0.14%	0.12%
4	0.00%	0.00%
5	0.00%	0.00%
Number of Orders	CTAS 4 Data	CTAS 4 Generated
1	5.94%	8.80%
2	1.23%	2.26%
3	0.10%	0.81%
4	0.00%	0.00%
5	0.00%	0.00%
Number of Orders	CTAS 5 Data	CTAS 5 Generated
1	3.83%	21.74%
2	1.09%	13.04%
3	5.46%	13.04%

4	0.00%	0.00%
5	0.00%	0.00%

Table 3.9: Comparison between data and generated admission and discharge rates.

Overall Data Discharge	Overall Data Admission	Overall Generated Discharge	Overall Generated Admission
87.25%	12.75%	87.84%	12.16%
CTAS 1 Data Discharge	CTAS 1 Data Admission	CTAS 1 Generated Discharge	CTAS 1 Generated Admission
46.21%	53.79%	53.15%	46.85%
CTAS 2 Data Discharge	CTAS 2 Data Admission	CTAS 2 Generated Discharge	CTAS 2 Generated Admission
81.26%	18.74%	82.30%	17.70%
CTAS 3 Data Discharge	CTAS 3 Data Admission	CTAS 3 Generated Discharge	CTAS 3 Generated Admission
92.92%	7.08%	92.02%	7.98%
CTAS 4 Data Discharge	CTAS 4 Data Admission	CTAS 4 Generated Discharge	CTAS 4 Generated Admission
97.20%	2.80%	96.85%	3.15%
CTAS 5 Data Discharge	CTAS 5 Data Admission	CTAS 5 Generated Discharge	CTAS 5 Generated Admission
97.81%	2.19%	96.85%	4.35%

These results where deemed acceptable by an on staff physician and the study proceeded to the modeling of the ED. At first glance some of the results for the

CTAS level 4 and 5 simulations seem to be very off, this is due to the fact that they represent an extremely low proportion of the data as can be seen in table 3.4.

3.4 Discussion

In this chapter the process through which patients for the simulation of the ED were generated was detailed. To begin the modeling of patient arrivals was done via a non-stationary Poisson process. With some slight adjustments this was able to effectively reproduce arrival patterns for the patients in the high acuity queue for both weekdays and weekends. Following this patient profiles where created. These used a combination of the patients chief complaint, sex, age and CTAS level. These where done with the first three influencing the CTAS level. With the use of support and confidence important relationships where able to be identified allowing for the generation of profiles to fit the patient demographic that visits the ED. Finally similar procedures where used to create the number of laboratory testing and imaging procedure rounds that the patients where required to go through. Validation information was then presented to the reader to illustrate the effectiveness of the technique.

This process could be performed for any given ED allowing for the testing of several schedule or policy changes. Additionally, since the data is synthetically generated multiple sets can be created ensuring that policy creation is not over fitting the data. This in particular, is a large strength the method has over simply using the gathered data for testing.

Chapter 4

Modeling the Emergency Department

4.1 Provided Data

As discussed in the prior chapter, the data is broken into two sections; ED data and laboratory and imaging data. Since the process of generating patients was established in the previous section, the next step in building the simulation is calculating the time spent waiting for events to occur.

4.2 Evolution of the Model Trough Iterations

The model was developed iteratively with continual feedback from an emergency physician to ensure the correct level of detail was incorporated in the simulation. It was decided that time would be measured in minutes because the data has no measurement that are at a finer level.

To begin, the first iteration of the model was a simplified representation of the system with a straight forward route that a patient would take through the ED (4.1). In this first iteration the patient will see the physician for an initial assessment, wait for any laboratory tests and imaging procedures to be completed, see the physician again for a reassessment and then be admitted or discharged. Patients are chosen for initial assessment by an accumulating priority queue that makes use of the patient's CTAS level. The conditions that allow a new patient to enter the initial assessment are that a bed is available (i.e., there are 50 total available) and a that a physician is available to see the patients. In this iteration, a physician works in two distinct phases; the first where they are accumulating new patients and the second where they are servicing their current patients.

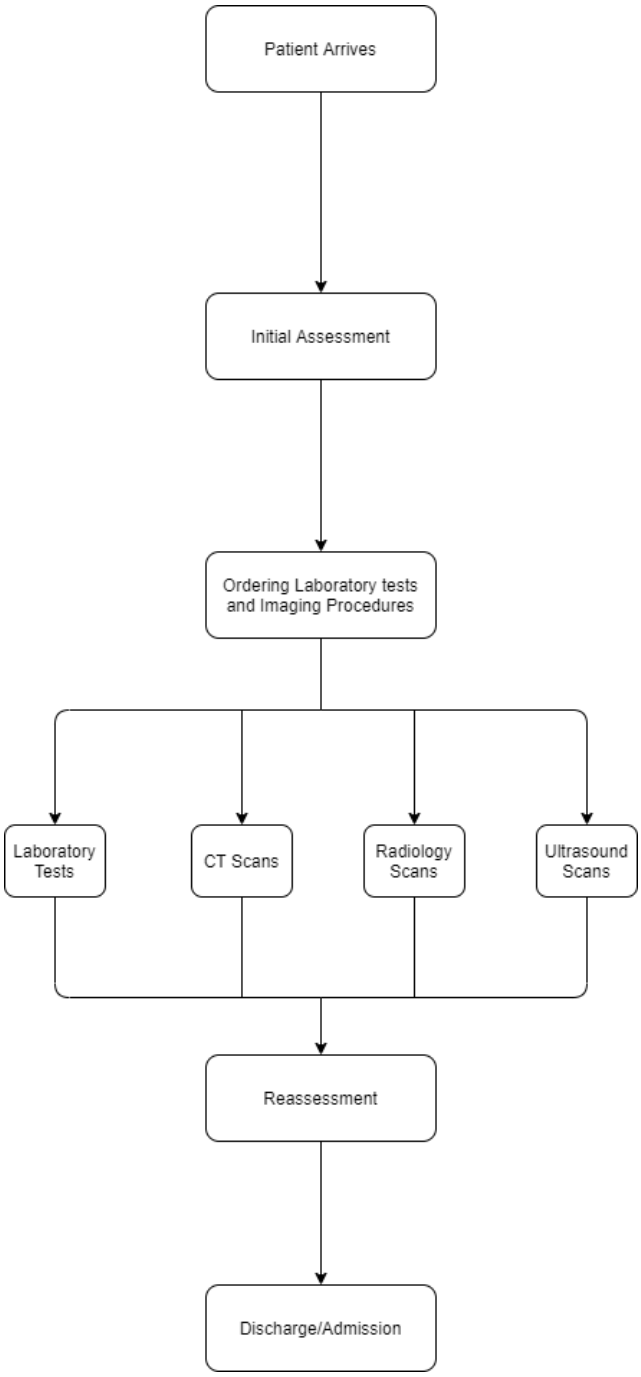

Figure 4.1: First Iteration of the ED model.

The above described model was further developed after discussion of the importance of the need to address multiple rounds of laboratory test and imaging procedures. An example of this expanded model can be seen in Figure 4.2. In the updated model, the patient will again wait for the laboratory tests and imaging procedures as

well as another reassessment to be completed before admission or discharge. Additionally, the added event of bed blocking was considered in the case of admission.

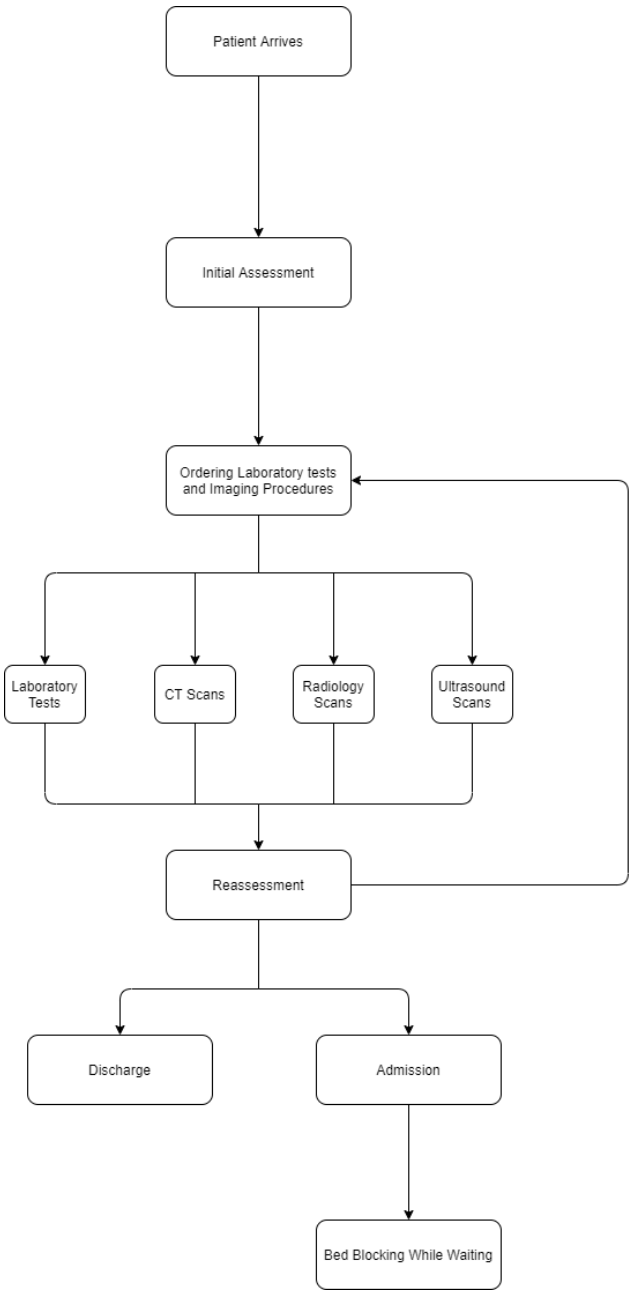

Figure 4.2: Second iteration of the model.

In the next set of iterations of the model, the event chain that a patient can follow did not change, however, physician and patient interaction was modified. After discussion it was established that the rapid accumulation of new patients does not

have a hard time limit in practice and in reality works with a soft constraint on the maximum number of patients per physician. In the simulation, the maximum number of patients that the physician can assess is 21 except for the overnight physician who does not have a limit. Additionally. the process of patient handover was added for those physicians who are at the end of there scheduled shift, although it will not interrupt the assessment that the physician is already in. It was also established that not all patients require the use of their beds throughout their stay and only for assessments. As it is not possible to obtain any information regarding this from the data it was decided that the patients that would likely need their beds for the entire stay would also likely be admitted, which was therefore used for determining this. Lastly it was decided that the time required to perform the imaging procedures was relatively small compared to the time required to wait for them to occur. Due to the fact that their is no information in the data of how long a patient was in the procedure, the patient only needs to undergo the waiting period for the procedure.

4.3 Modeling of Individual Steps

With the flow of the model established, it was next determined how patients were chosen for assessment and the time for each event within the patients stay. Data timestamps for the various stages of the patients stay in the ED were compared to ensure that the chronological order was logical as a method for checking for errors in the data.

4.3.1 Choosing Which Patients are Served First

To determine which patients were chosen for assessment, a priority queue was used. The aging of priority values and their initial values can be seen in figure 4.3 and table 4.1. These values where estimated through a process of trial and error in an effort to get the closest simulated CTAS level mean PIA and LOS times to the actual data. There is no data available to estimate this process. In practice, it is up to each physician to determine patients priority and it is not a structured set of rules, but instead a consideration of CTAS level, patient waiting time and patient factors such as age and past medical history. The comparison of simulated results with the actual PIA and LOS in the data can be seen in the validation section of this chapter.

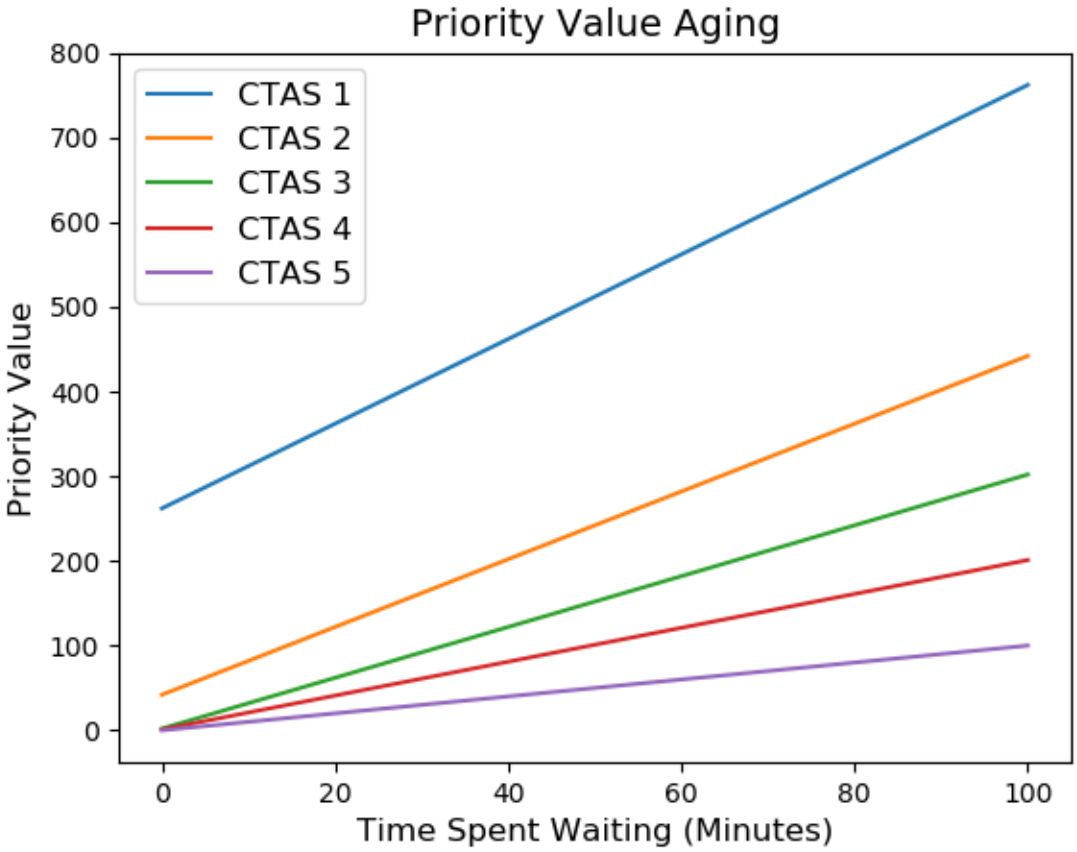

Figure 4.3: Evolution in patient priority values over time.

Table 4.1: How priority levels for patients evolve over time.

CTAS Level	Starting Priority Value	Priority Increase Per Minute
1	262	5
2	42	4
3	2	3
4	1	2
5	0	1

4.3.2 Modeling the Time Spent With the Physician

For the time a physician spends with patients during assessments, ordering writing, reviewing results of investigations and subsequent reassessments, there is unfortu-

nately no data availabe. Therefore, the results from a previous study from Ontario that investigated the time spent with patients by CTAS level was used [9]. This study allows us a starting point for determining times for the simulation. The distributions shown in the study appear to be exponential and therefore that is what was used for the simulations. The mean time spent with each patient based on CTAS level, appear to be high for our purposes due to the high volume and higher proportion of higher acuity patients. The mean times were adjusted and can be seen in table 4.2. They were adjusted through a process of trial and error with some insight from a physician on staff at the ED.

Table 4.2: The amount of time patient's spend with physicians.

CTAS Level	Old Mean (Minutes)	New Mean (Minutes)
1	73.6	59.1
2	38.9	29.8
3	26.3	18.6
4	15.0	11.0
5	10.9	7.7

With the mean times spent with patients based on CTAS level established the question remains how is it divided in the patient's stay. For the purpose of simplicity three types of assessment are established; initial assessment, reassessment and repeated reassessment. A genetic algorithm was made as method of establishing the breakdown of the time. The algorithm uses the generated patients to determine a multiplication factor for each CTAS level and each type of assessment. The resulting factors are shown in table 4.3. The algorithm uses the bounds shown in table 4.4 to generate acceptable sets of multiplication factors for each CTAS level. The repeated reassessment bounds are constrianed far lower than the others due to the fact that the vast majority of the repeated reassessments are due to laboratory test and would only need a small amount of the physicians time to get an update on the patient's status.

Table 4.3: Generated breakdown of how much time a patient spends with a physician, during which part of their visit to the ED.

CTAS Level	Stage	Factor
1	Initial Assessment	0.43
1	Reassessment	0.54
1	Repeated Reassessment	0.10
2	Initial Assessment	0.57
2	Reassessment	0.54
2	Repeated Reassessment	0.10
3	Initial Assessment	0.81
3	Reassessment	0.27
3	Repeated Reassessment	0.11
4	Initial Assessment	0.82
4	Reassessment	0.36
4	Repeated Reassessment	0.09
5	Initial Assessment	0.78
5	Reassessment	0.39
5	Repeated Reassessment	0.13

Table 4.4: Bounds used in the genetic algorithm to generate the mean times patient's spend with physicians.

CTAS Level	Stage	Lower Bound (Minutes)	Upper Bound (Minutes)
1	Initial Assessment	30%	reassessment time
1	Reassessment	30%	80%
1	Repeated Reassessment	1%	20%
2	Initial Assessment	30%	80%
2	Reassessment	30%	initial assessment time
2	Repeated Reassessment	1%	20%
3	Initial Assessment	30%	90%
3	Reassessment	30%	40%
3	Repeated Reassessment	1%	20%
4	Initial Assessment	30%	95%
4	Reassessment	30%	initial assessment time

4	Repeated Reassessment	1%	20%
5	Initial Assessment	30%	95%
5	Reassessment	20%	initial assessment time
5	Repeated Reassessment	1%	20%

The resulting exponential distributions that are used in the simulation can be seen in figures 4.4, 4.5 and 4.6. These distributions are trimmed to negate tail effects. The trimming was done through trial and error to obtain the closed PIA and LOS values possible, this will be discussed further in the validation portion of this chapter. The lower and upper bounds for these distributions can be seen in table 4.5.

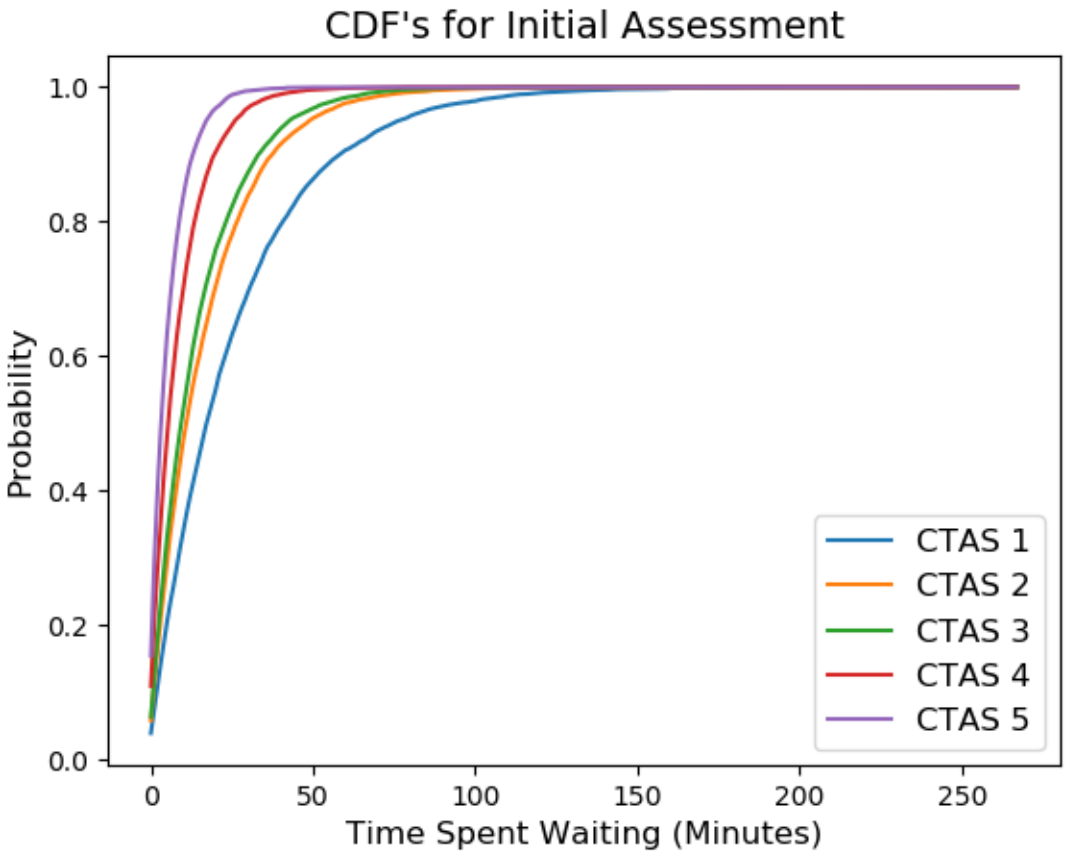

Figure 4.4: CDF's of time spent in initial assessments for CTAS levels.

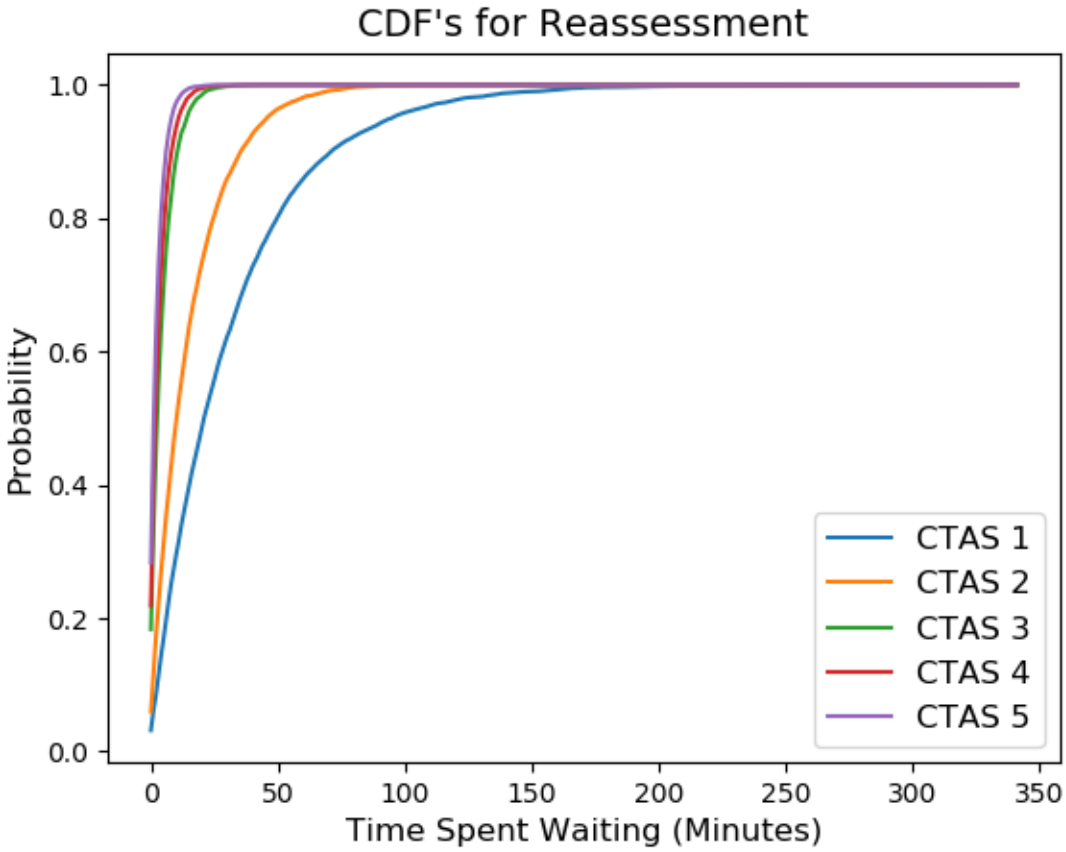

Figure 4.5: CDF's of time spent in reassessments for CTAS levels.

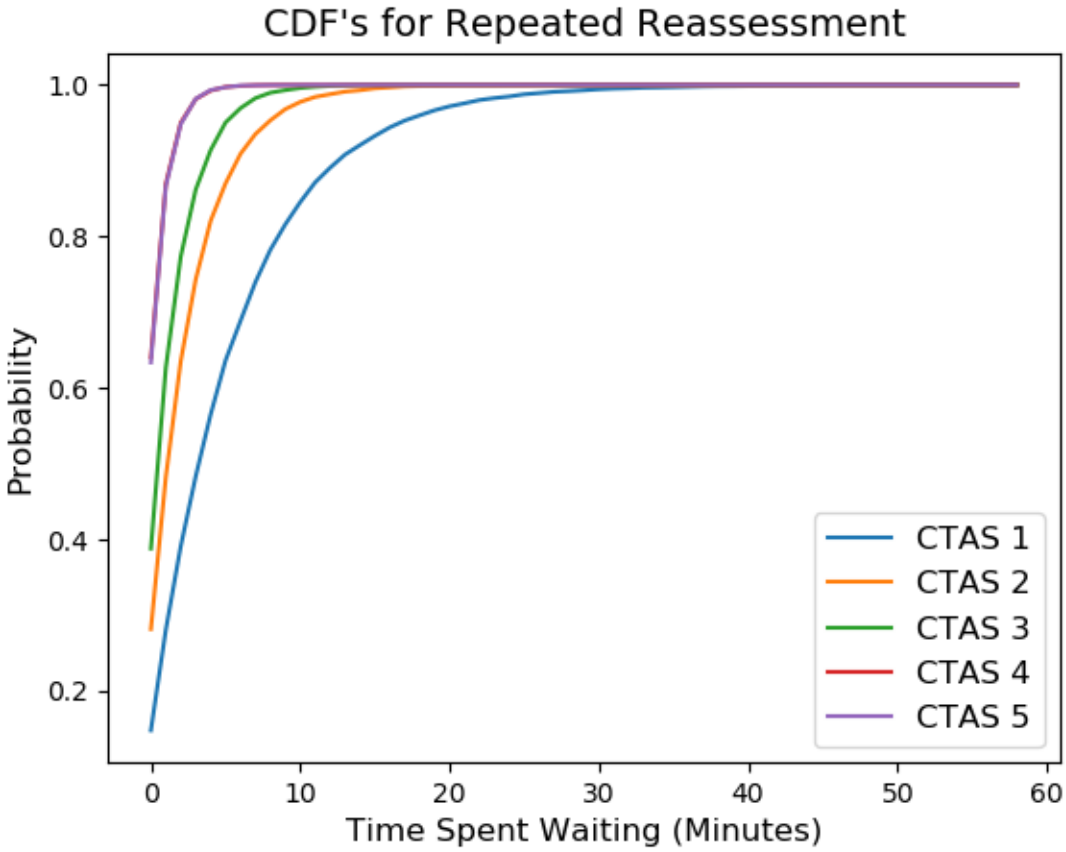

Figure 4.6: CDF's of time spent in repeated reassessments for CTAS levels.

Table 4.5: Bounds placed on the distributions of how much time patient's spend with physicians in order to avoid edge effects.

CTAS Level	Stage	Lower Bound (Minutes)	Upper Bound (Minutes)
1	Initial Assessment	9	59
1	Reassessment	11	73
1	Repeated Reassessment	2	17
2	Initial Assessment	6	39
2	Reassessment	5	30
2	Repeated Reassessment	1	9
3	Initial Assessment	5	34
3	Reassessment	1	14
3	Repeated Reassessment	1	5

4	Initial Assessment	3	20
4	Reassessment	1	11
4	Repeated Reassessment	1	3
5	Initial Assessment	2	13
5	Reassessment	1	8
5	Repeated Reassessment	1	2

4.3.3 Modeling the service time for laboratory tests and imaging procedures

For the laboratory tests and imaging procedures the data was grouped in a similar manner to the analysis of the number of sequential tests being done except the patient descriptors were not considered. When analyzing the data there was no noticeable difference between the CTAS levels in the waiting times. It was also no difference between weekends and weekdays. Since there are multiple time lengths in these groups the longest is used, as it was mentioned before often additional orders are added on while waiting. For each of these events three types of distributions where chosen for investigation; exponential, log normal and gamma. These were chosen as they are all common distributions used in queueing systems and simulation modelling.

Laboratory Tests

For laboratory tests there is a special consideration. Within the data there is information on time spent waiting for collection of samples from the patient and the time spent waiting for the samples to be processed (i.e., time spent waiting for the results). The question becomes whether to model these separately or as one event. The distributions considered for time spent waiting for collection can be seen in Figure 4.7. The waiting times for sample processing are shown in Figure 4.8. Finally, the times for the combination of both are shown in Figure 4.9. The results show that the best fitting distribution for time until collection is log normal, although it is closer to the exponential than the gamma. For the time until lab testing is completed, the best fitting distribution is the log normal, but it is closer to the gamma than the exponential. For this reason while log normal is likely the best fitting distribution for the combined laboratory testing times it does not fit as closely to the other two

distributions. However, upon discussion it was deemed acceptable and is used in the simulation. The lower bound is 29 minutes and the upper bound is 175 minutes.

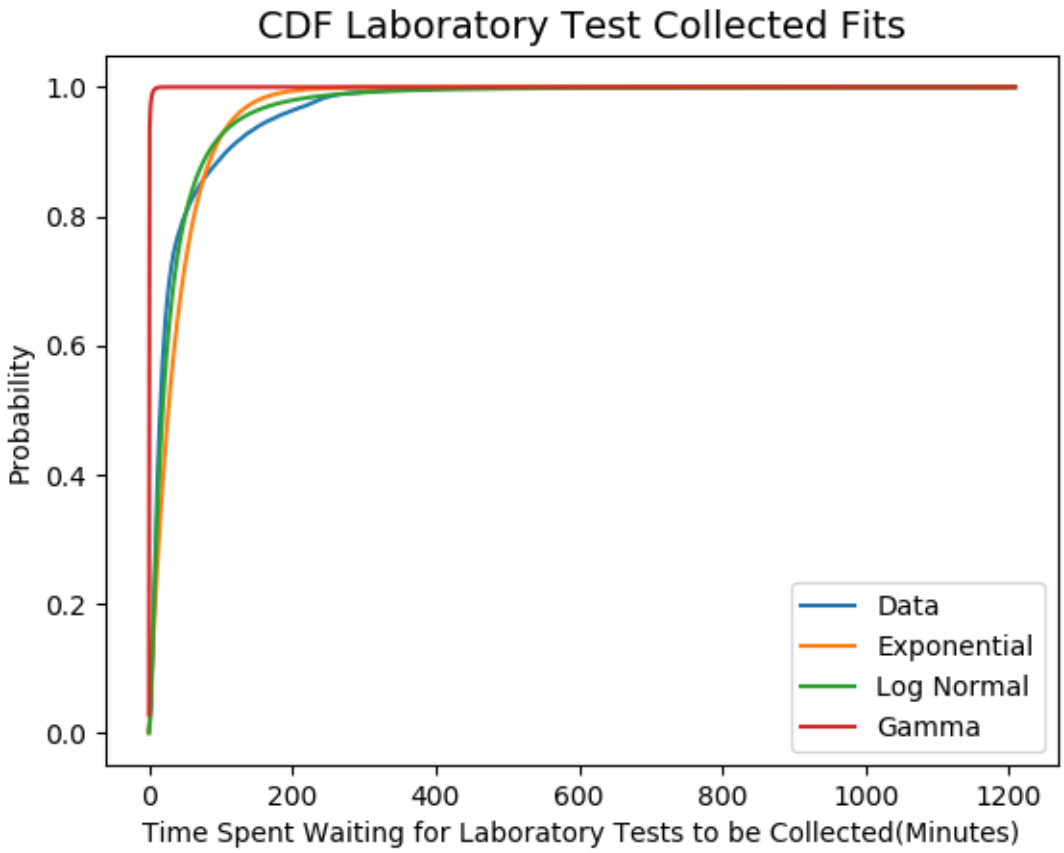

Figure 4.7: CDF's of time spent waiting for laboratory samples to be collected.

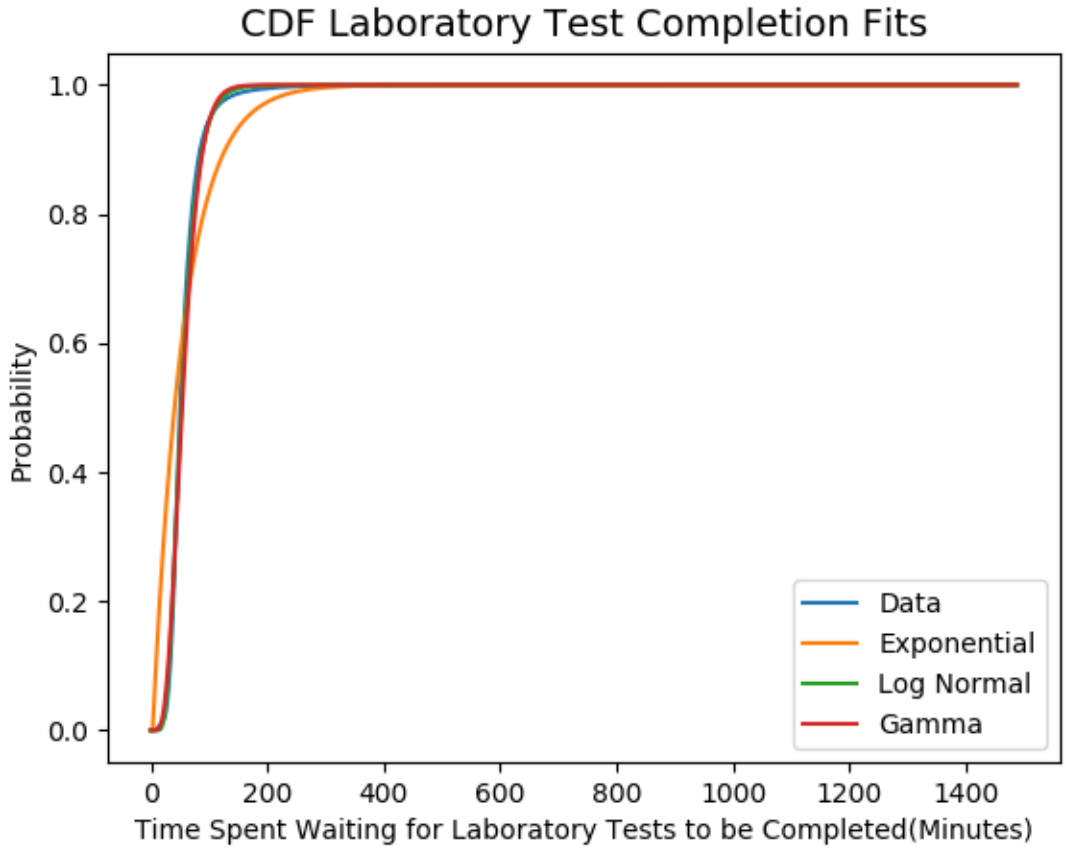

Figure 4.8: CDF's of time spent waiting for laboratory tests to be completed.

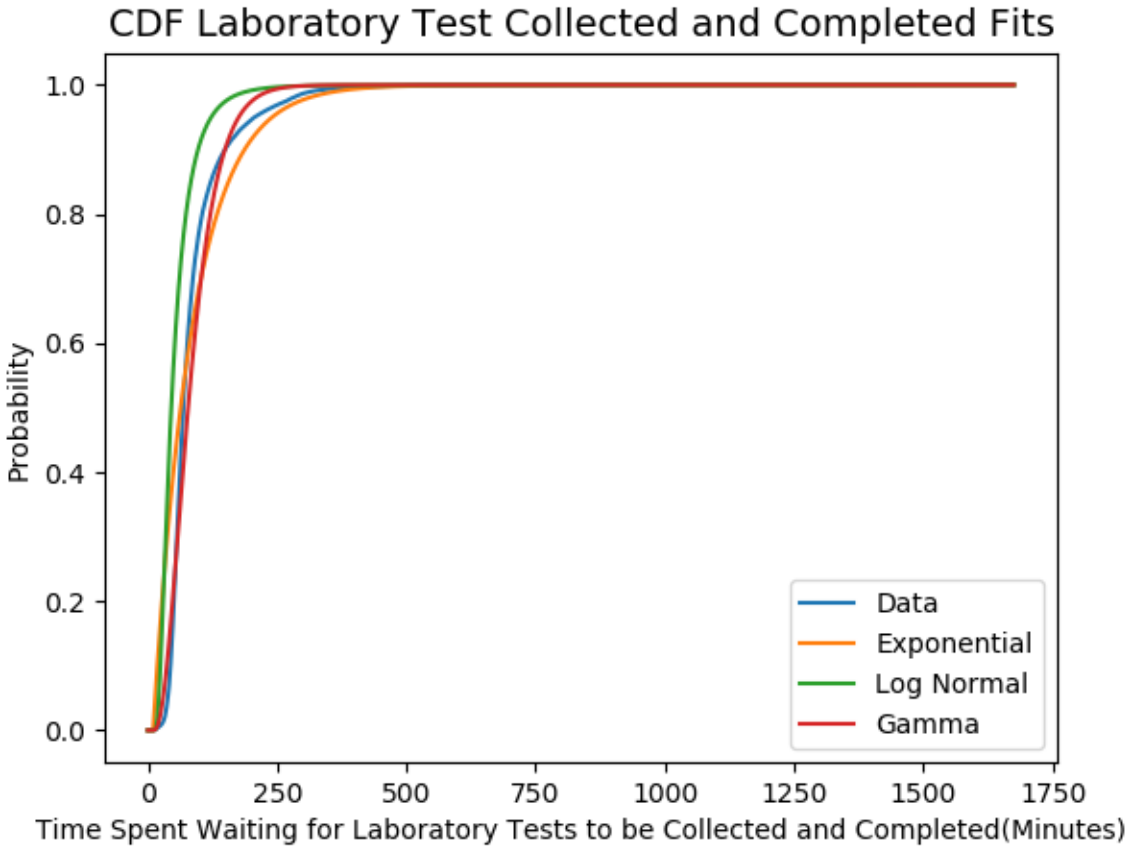

Figure 4.9: CDF's of time spent waiting for laboratory samples to be collected and the tests completed.

CT Imaging

For CT the scanner, there is no technician available from 0:00-7:00 except for emergencies. This produces variances in the distributions based on time. Through multiple groupings of the data based on the 24 bins (i.e., starting on the hour), they were grouped into four groups. These groups are as follows:

The group for 0:00-1:59 and 6:00-7:59 can be seen in figure 4.10. The distribution that fits best for this is the gamma.

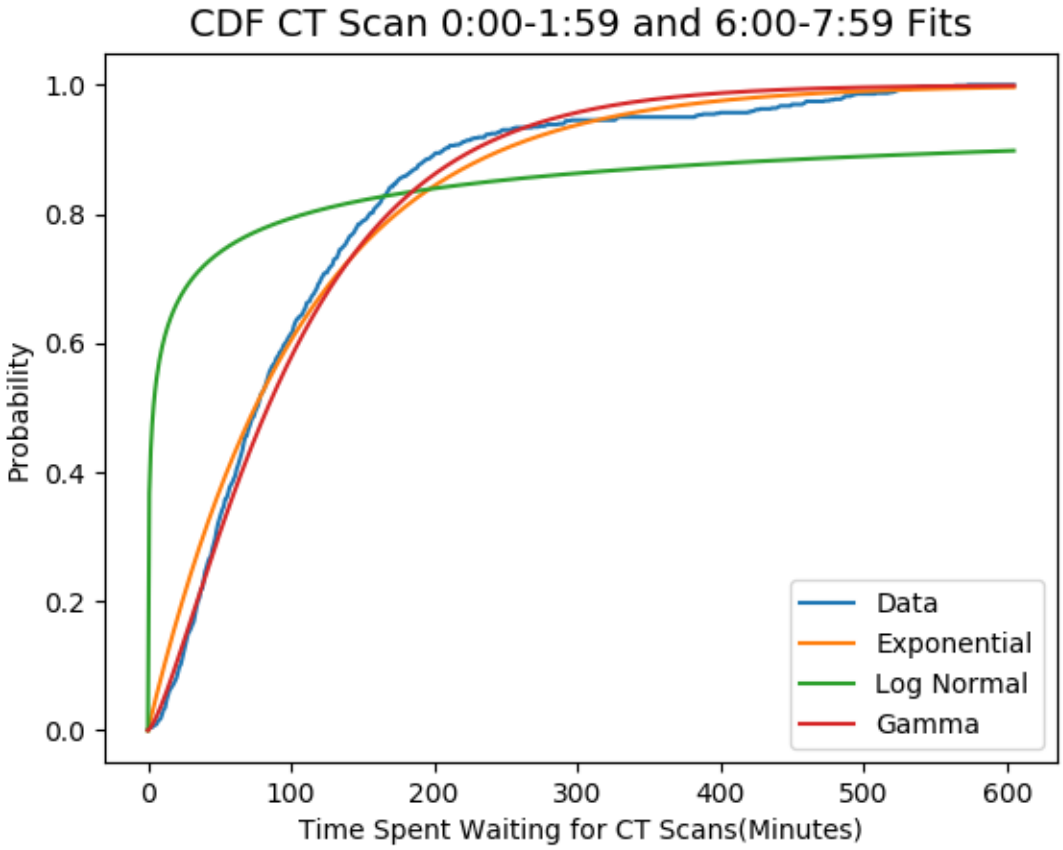

Figure 4.10: CDF's for CT scans ordered between 0:00-1:59 and 6:00-7:59.

The group for the 2:00-5:59 can be seen in figure 4.11. The distribution that fits best is the exponential.

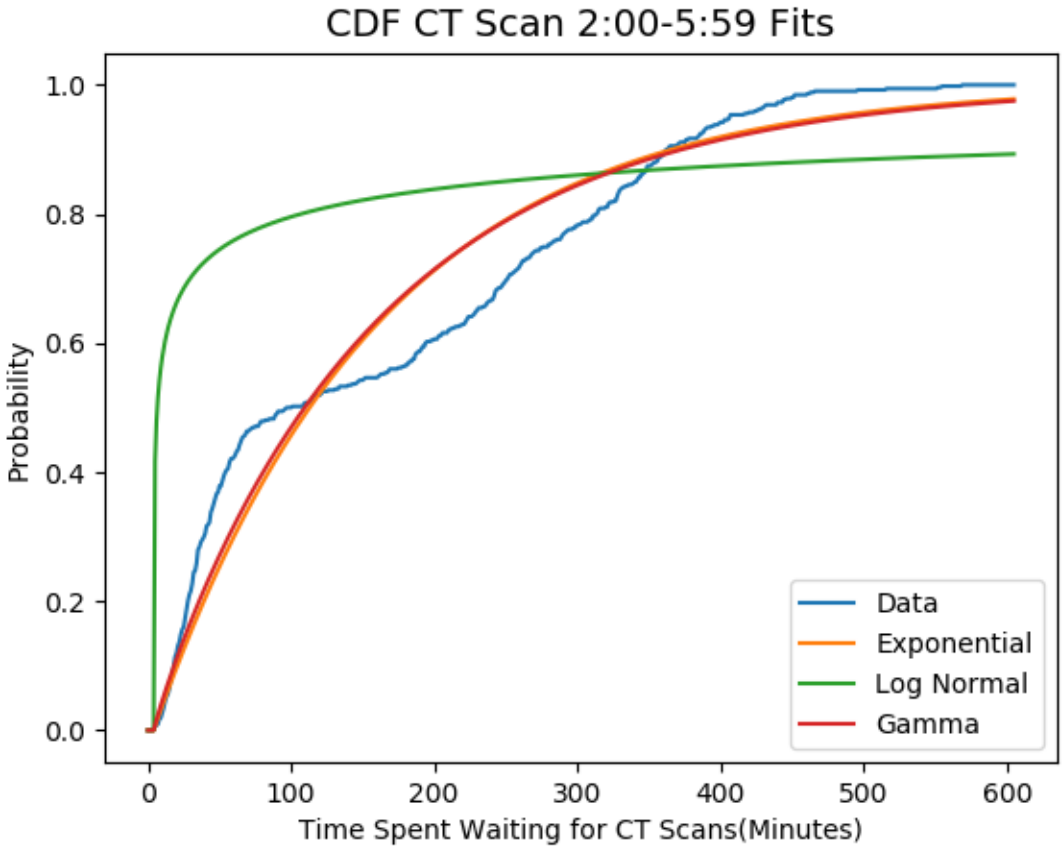

Figure 4.11: CDF's for CT scans ordered between 2:00-5:59.

The group for the 8:00-19:59 can be seen in figure 4.12. The distribution that fits best is the gamma.

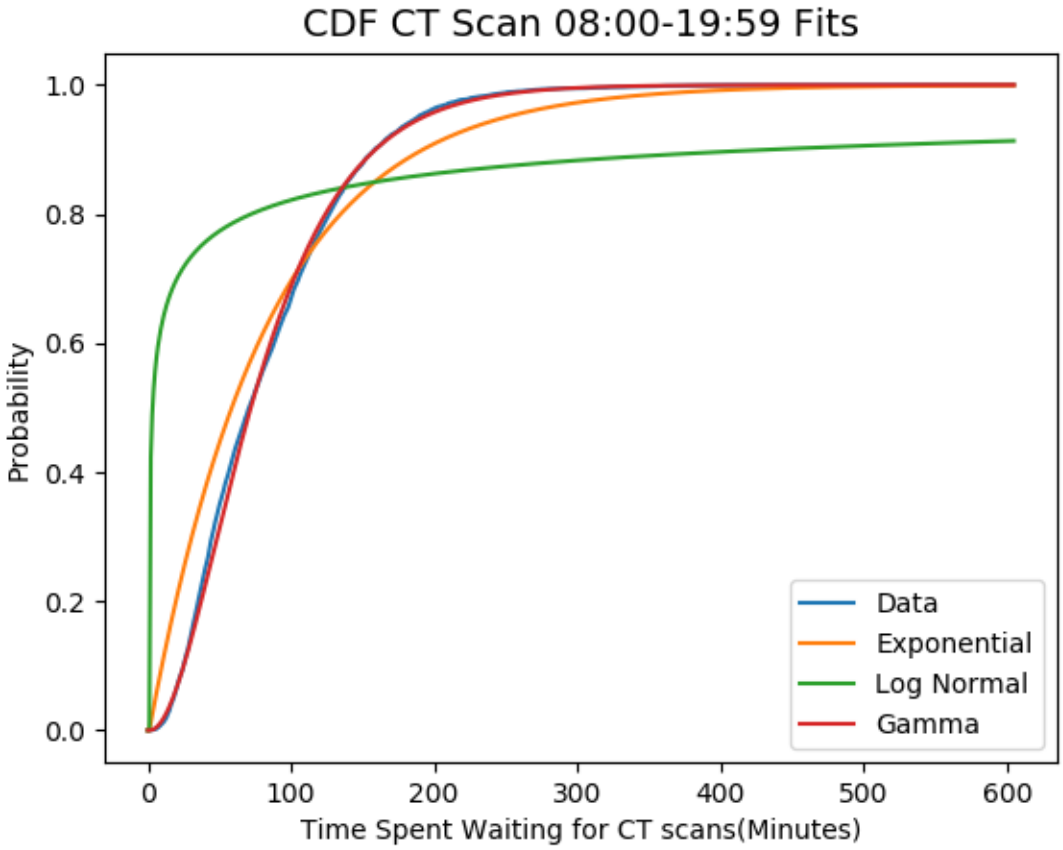

Figure 4.12: CDF's for CT scans ordered between 8:00-19:59

The group for the 20:00-23:59 can be seen in figure 4.13. The distribution that fits best is the gamma.

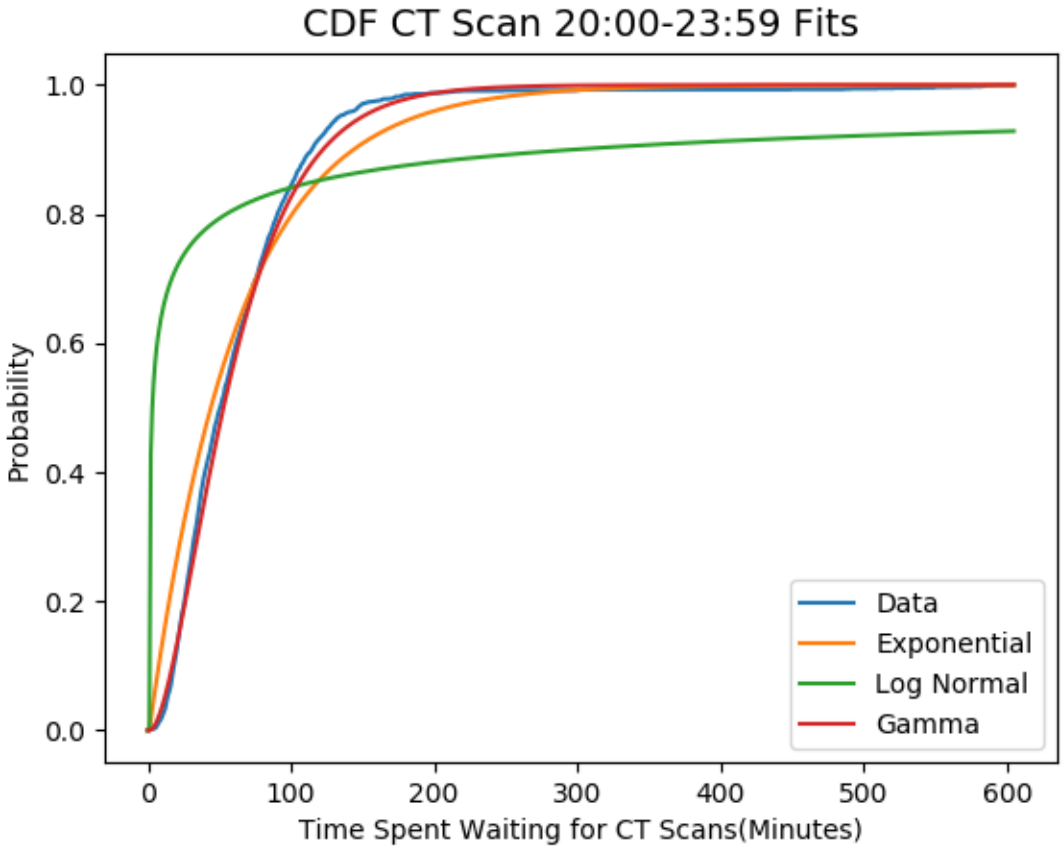

Figure 4.13: CDF's for CT scans ordered between 20:00-23:59.

As with the distributions for the time spent with physicians these distributions are all truncated to negate tail effects. The upper and lower bounds for the chosen distributions are shown in Table 4.6.

Table 4.6: Bounds used for CT scan wait time distributions to avoid edge effects.

Times CT is Ordered	Lower Bound	Upper Bound
0:00-1:59 and 6:00-7:59	11	286
2:00-5:59	12	470
8:00-19:59	17	192
20:00-23:59	11	149

Radiology Imaging

For the time spent waiting to complete imaging, the distributions can be seen in Figure 4.14. The distribution with the best fit is the exponential distribution. Again, these have been truncated to negate tail effects. The lower bound is 3 minutes and the upper bound is 182 minutes.

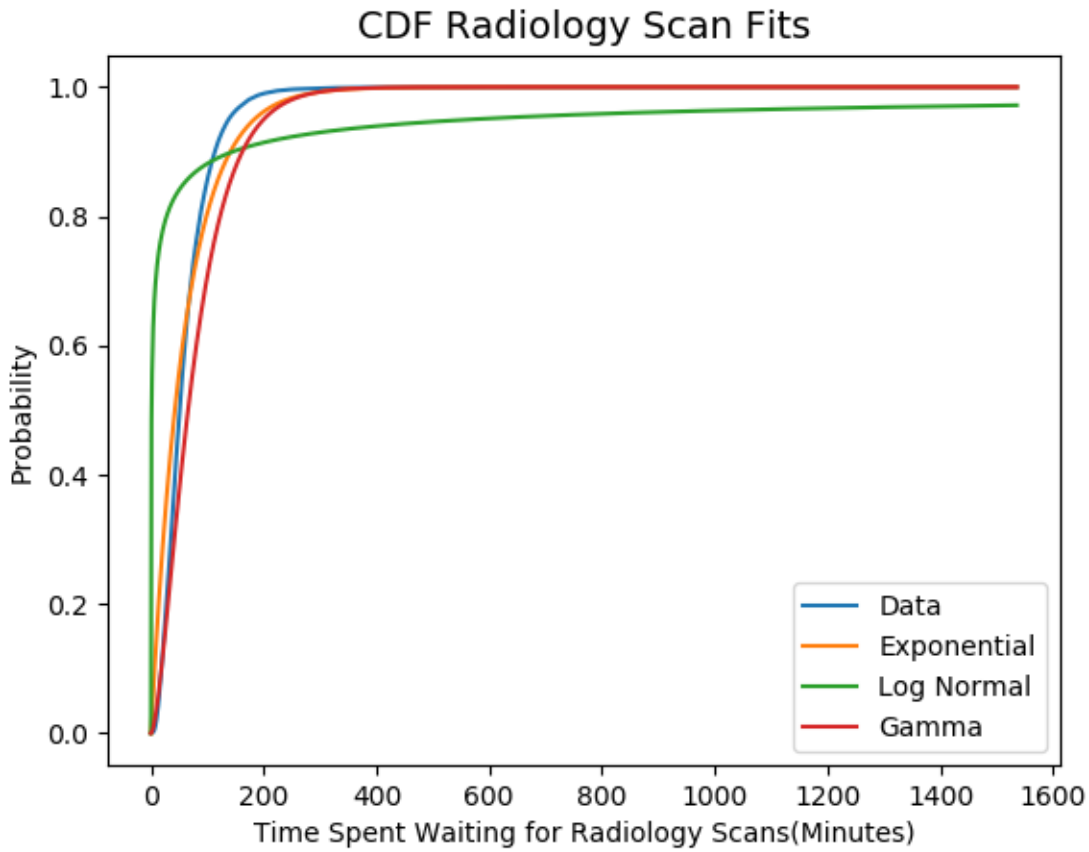

Figure 4.14: CDF's for radiology scans ordered.

Ultrasound Imaging

The ultrasound imaging, like the CT imaging, relies on technicians that are not always scheduled (i.e., 23:00-7:00). The same process was used to produce the groupings for the distributions. The groups are as follows:

The group for the 0:00-3:59 can be seen in figure 4.15. The distribution that fits best is the gamma.

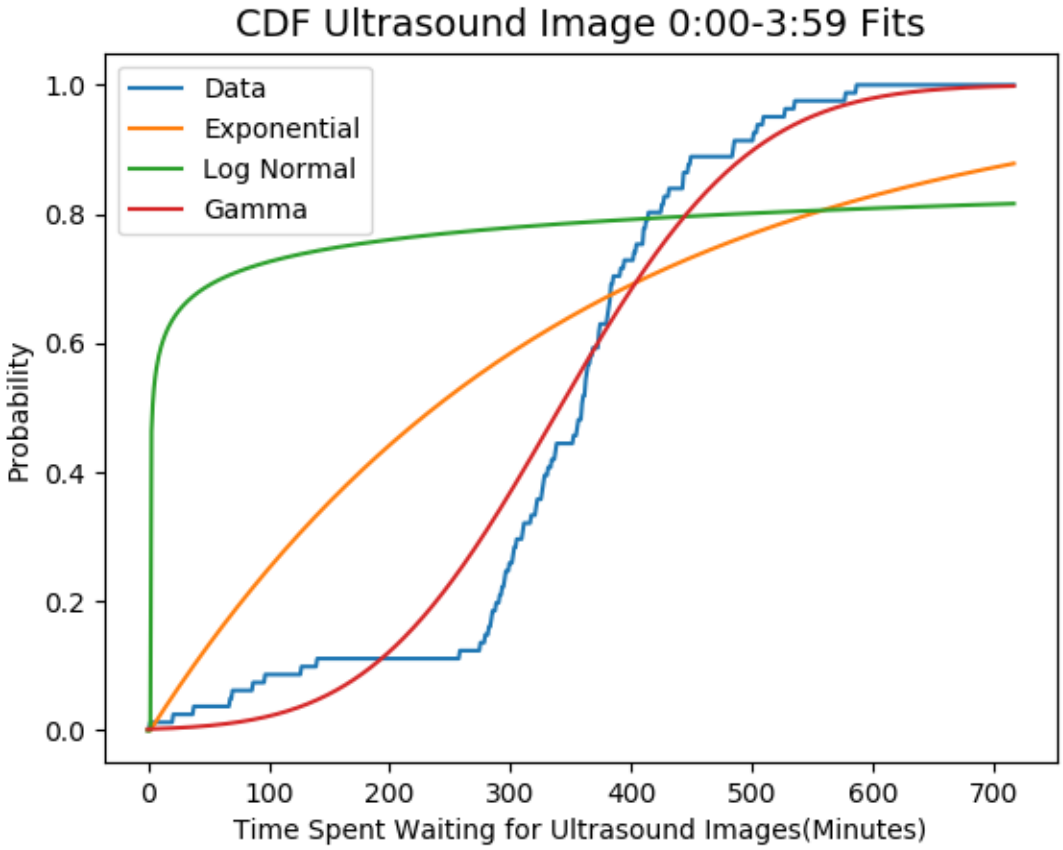

Figure 4.15: CDF's for US scans ordered between 0:00-3:59.

The group for the 4:00-7:59 can be seen in figure 4.16. The distribution that fits best is the gamma.

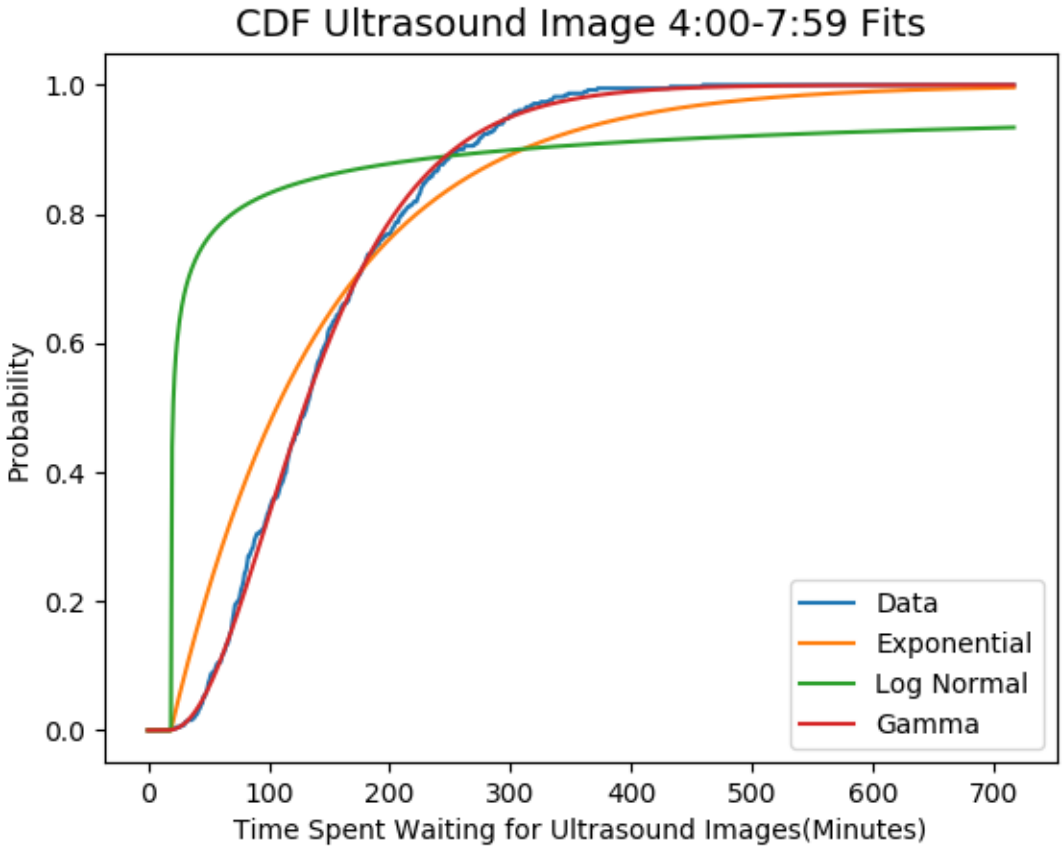

Figure 4.16: CDF's for US scans ordered between 4:00-7:59.

The group for the 8:00-19:59 can be seen in figure 4.17. The distribution that fits best is the gamma.

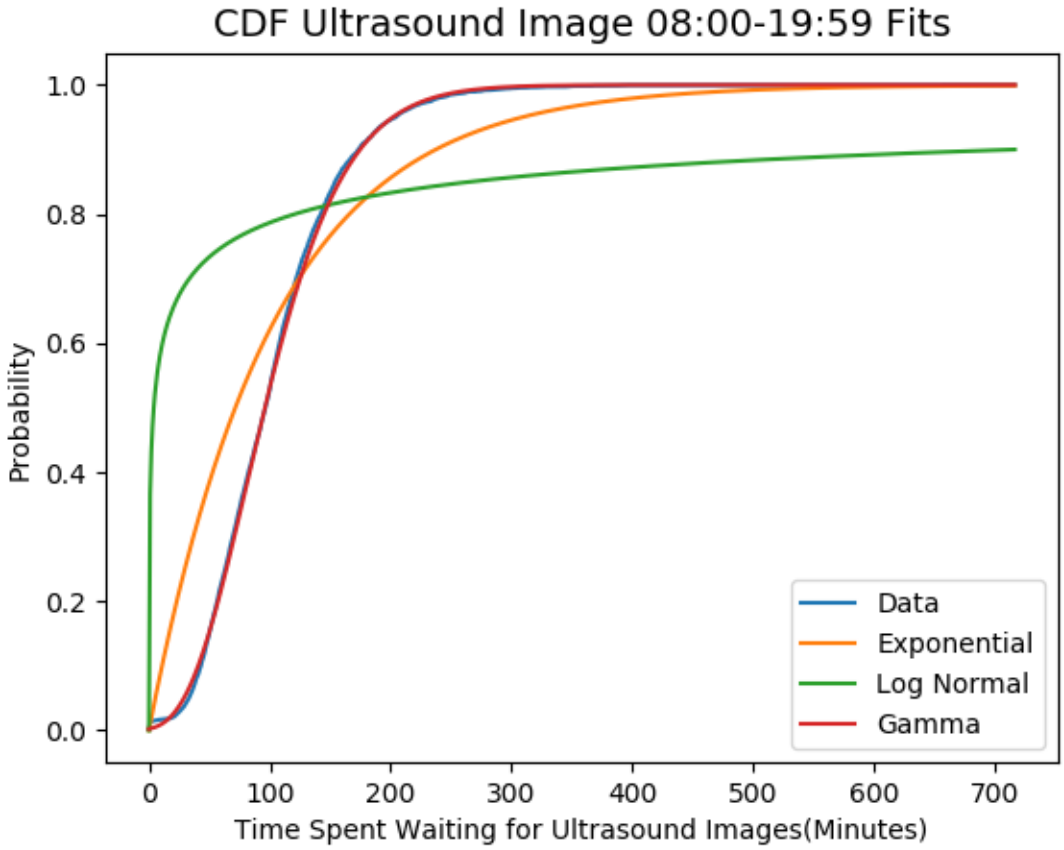

Figure 4.17: CDF's for US scans ordered between 8:00-19:59.

The group for the 20:00-23:59 can be seen in figure 4.18. The distribution that fits best is the gamma.

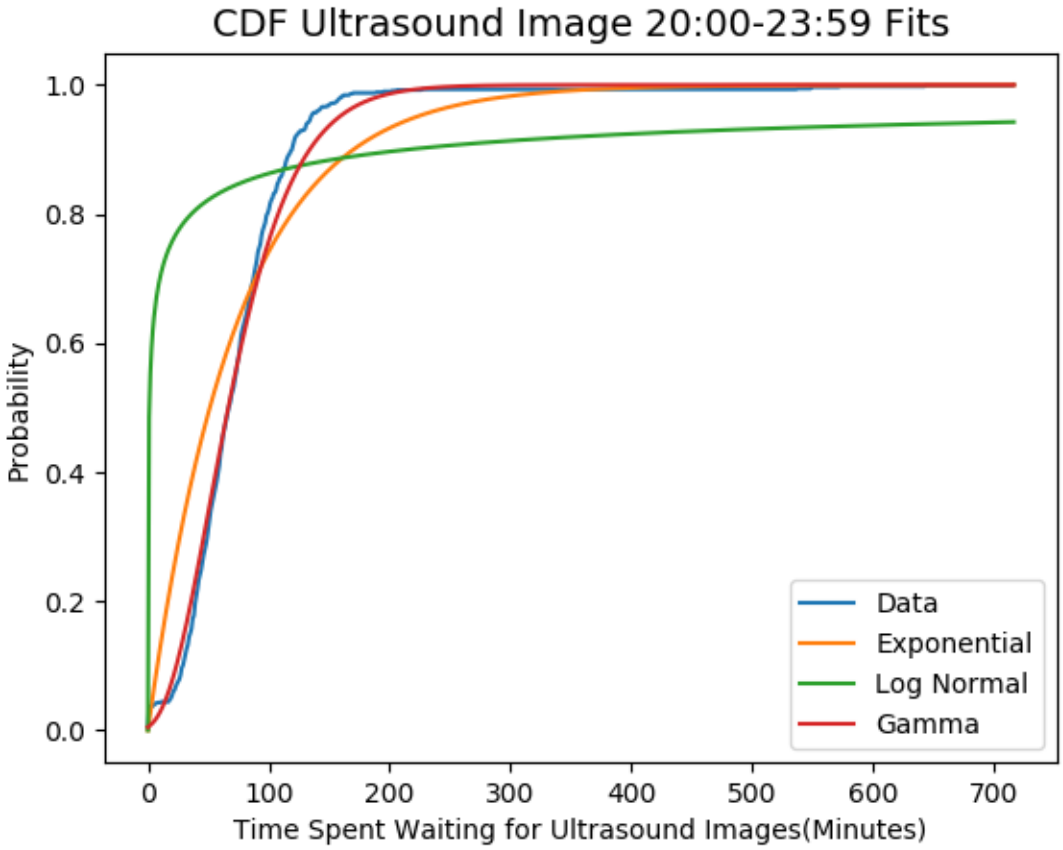

Figure 4.18: CDF's for US scans ordered between 20:00-23:59.

These distributions are again truncated to negate tail effects and the bounds can be seen in Table 4.7.

Table 4.7: Bounds used for ultrasound wait time distributions to avoid edge effects.

Time US is ordered	Lower Bound	Upper Bound
0:00-3:59	142	547
4:00-7:59	45	299
8:00-19:59	28	201
20:00-23:59	45	106

4.3.4 Modeling the Time Spent Bed-Blocking

For the time a patient spends bed-blocking (i.e., waiting in the ED for transfer to a hospital floor) the same three distributions were investigated. It was found that there was no distinct difference in distributions between weekdays and weekends. Similarly there was no differences between CTAS levels. The distributions can be seen in Figure 4.19. The distribution that fits best is the lognormal. Again this is tuncated to negate tail effects. The lower bound used is 23 minutes and the upper bound is 1896 minutes.

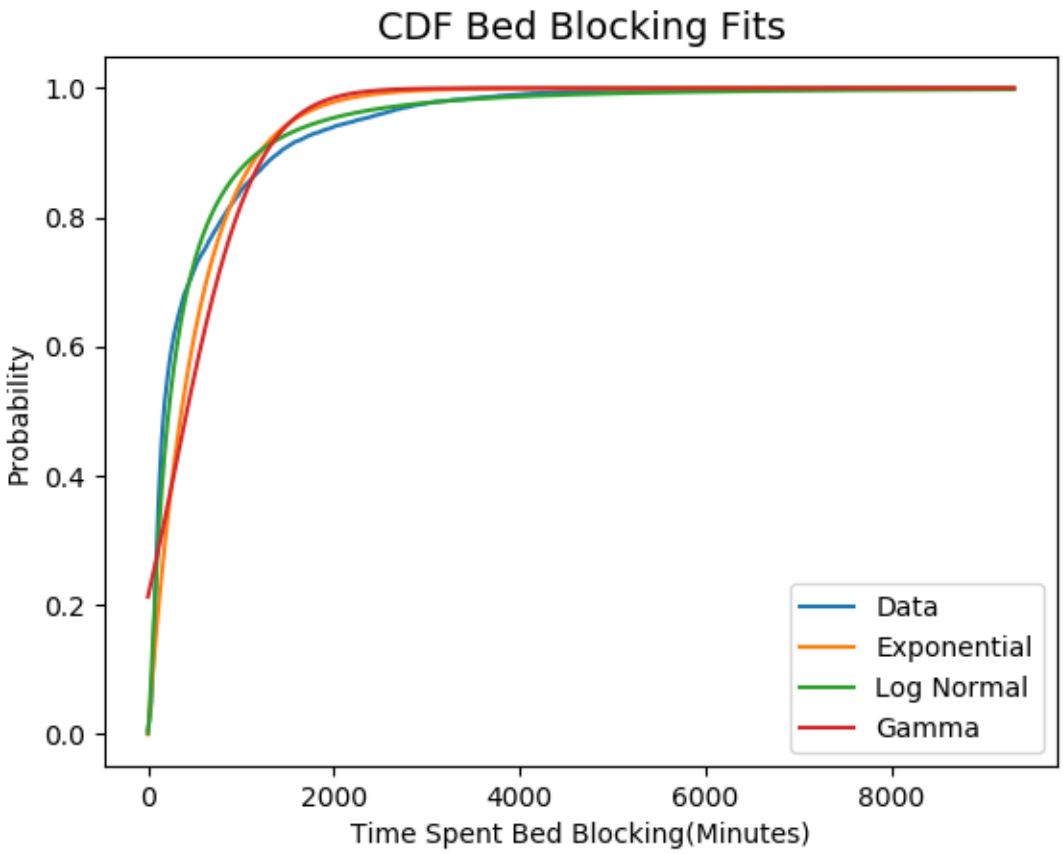

Figure 4.19: CDF's for bed blocking.

4.4 Validation of the Model

The process of validating the model of the ED was done by using the physician schedule that was in place during data collection and simulating 365 days of patients and then comparing the metrics PIA and LOS to those from the data.

While funding only considers how many patients meet target PIA and LOS times for a particular CTAS level. For the purposes of validation we will be looking at the metrics on patients as a whole and in the individual CTAS levels. It should also be noted that the results from patients in the first day are neglected as the model startup period could have effects. As well, the last day of data is also ignored since the patients may not complete their visit . Therefore these function as warm up and cool down periods, respectively. Tables showing the comparison of the PIA and LOS times can be found in Table 4.8 and 4.9 respectively. While graphical representations can be found in Figures 4.20 and 4.21. In the figures the bounds of the boxes represent the 25th and 75th percentile. While the whiskers represent the 5th and 95th percentile.

Table 4.8: Comparison between data and simulated PIA.

Type	Data Mean(Minutes)	C.I. 95%	Sim Mean(Minutes)	C.I. 95%
Overall	77.86	77.23-78.49	84.83	84.30-85.36
CTAS 1	21.38	19.62-23.13	24.66	22.76-26.57
CTAS 2	62.36	61.34-63.37	66.73	66.06-67.39
CTAS 3	91.02	90.18-91.85	95.89	95.17-96.60
CTAS 4	83.67	80.58-86.76	137.99	132.33-143.65
CTAS 5	83.01	72.85-93.17	279.43	162.80-396.07

Table 4.9: Comparison between data and simulated LOS.

Type	Data Mean(Minutes)	C.I. 95%	Sim Mean(Minutes)	C.I. 95%
Overall	299.55	297.89-301.22	293.73	292.34-295.12
CTAS 1	346.23	335.06-357.39	320.95	309.39-332.51
CTAS 2	331.73	328.77-334.69	295.78	293.56-298.00
CTAS 3	281.31	279.26-283.36	291.45	289.65-293.26
CTAS 4	214.69	207.67-221.71	295.65	283.99-307.30
CTAS 5	190.20	169.75-210.64	452.09	316.60-587.58

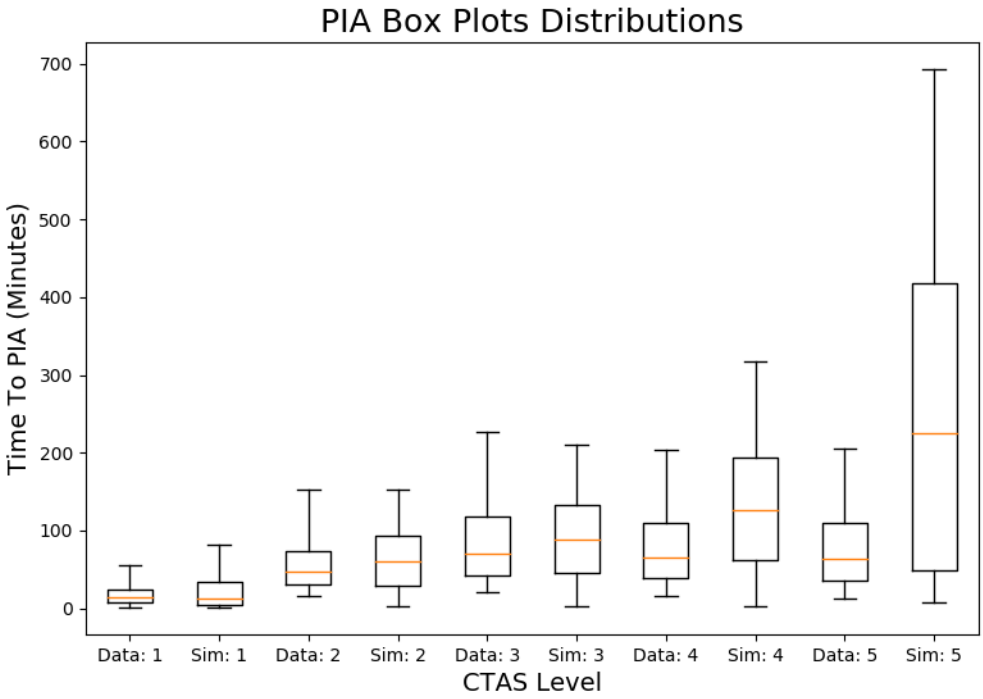

Figure 4.20: Boxplots for comparing PIA of data and simulation.

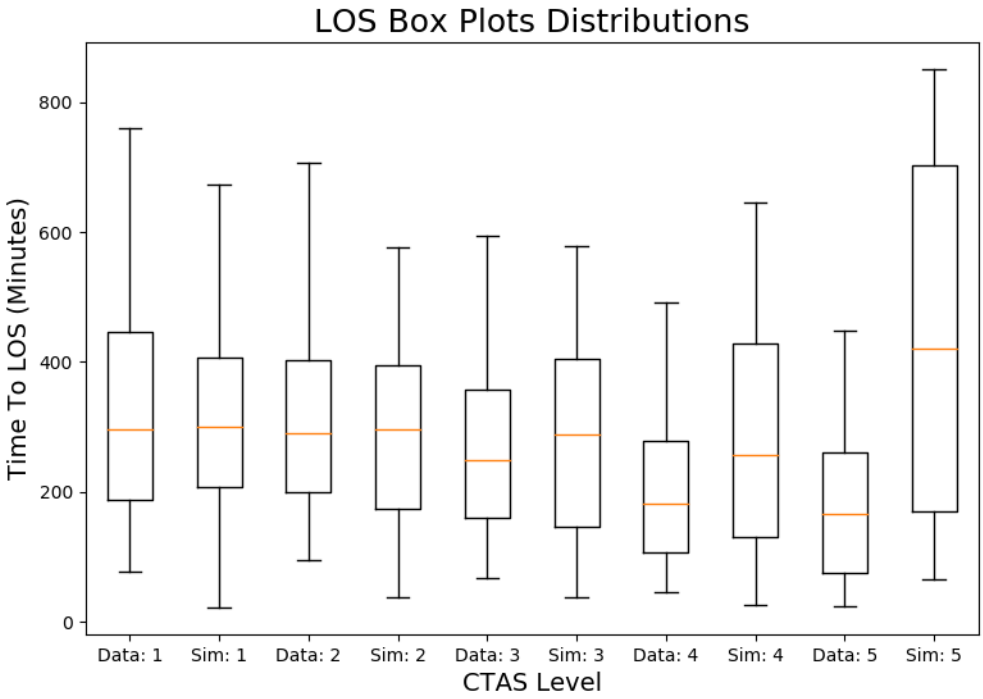

Figure 4.21: Boxplots for comparing LOS of data and simulation.

These results were deemed acceptable by an ED staff physician and the study proceeded to finding the optimal candidate schedule. Although, the result for CTAS 4 and 5 simulated patients does not represent the actual data well, these patients make up a very small portion of the data and are therefore difficult to model appropriately. In the previous chapter in Table 3.4, it can be seen that the these two levels makeup 3.83% of the data and 2.07% of the generated patients. Furthermore, the high acuity queue is not designed for these patients and they should be sent to the fast track queue. None the less, it appears that some make their way into the queue.

4.5 Discussion

In this chapter the construction of the process used to evaluate schedules is illustrated. To begin the patients stay was broken up into events that a patient goes through during they're stay. When put together this forms a series of queues patients proceed through while occupying resources. Individual distributions were produced modeling the time patients spend waiting for laboratory testing, imaging procedures and bed

blocking. For these several approaches were considered before deciding upon the final ones. The time spent with physicians was modeled based on prior work with some adaptions based on trial and error and physician input to fit the ED at the TBRHSC. This was done due to a lack of information in the data in regards to the time spent with physicians. The method for determining which patient is seen next by physicians uses an accumulating priority queue. The aging of the priority values was determined using trial and error with physician input as no information regarding it was present in the data and no prior works on the topic could be found. Validation results are then presented to the reader to prove the validity of the method.

This model could be easily replicated to match of EDs allowing for optimization of their physician schedules. In addition, with slight modifications it could be used to test policy changes that were discussed in the related work section that have not been considered at the TBRHSC. Also, it could be used in cost benefit analysis for additional staff and resources. Alternatively, due to the separation between the method and the patient data it could be used to forecast expected metrics if a change in patient demographics is expected. Therefore this could be an extremely useful tool for ED management.

Chapter 5

Cluster Partitioning

The secondary topic of investigation in this study was the cluster partitioning problem. This is commonly referred to as a minimum cut or maximum cut problem depending on the objective, our focus was the minimum cut. To begin, a graph is presented to describe the problem in question. For our purposes we will be looking at undirected graphs. The objective of the problem is to separate the graph into a specified number of n partitions. These partitions may be capacitated or uncapacitated, for our purposes they are capacitated. The minimum cut is the partitioning of the graph in which the set of edges that must be removed to separate these partitions into separate graphs is the least total weight possible. The max cut is the partitioning in which it is the most total weight possible that is removed from the graph to separate the partitions.

5.1 Integer Linear Programming

In order to assess the performance of the developed algorithms, integer linear programming (ILP) was used. In ILP, a problem is represented by a space specified by parameters and variables. This space is then restricted to a smaller subset through the use of constraints. The solver is direct through the use of an objective function in it's search for the optimal solution. As mentioned above we are considering the minimum cut problem.

The variables

Parameters and their descriptions are as follows:

V: The set of vertices

S: The set of partitions

s_i: The capacity of partition i

The variables and their descriptions are as follows:

$\omega(u,v)$: represents the weight of the edge between the vertices u and v

$y^i_{u,v}$: represents whether both the vertices u and v are within partition i

x^i_u: represents the presence of vertex u in partition i

The equations that govern the minimum cut problem can be seen in equations 5.1, 5.2, 5.3, 5.4, 5.5 and 5.6.

$$minimize \sum_i \sum_{u,v} \omega(u,v) y^i_{u,v} \tag{5.1}$$

$$\sum_i x^i_u = 1, \forall u \tag{5.2}$$

$$y^i_{u,v} = |x^i_u - x^i_v|, \forall u, v, i \tag{5.3}$$

$$\sum_u x^i_u = s_i, \forall i \tag{5.4}$$

$$x^i_u \in 0, 1 \tag{5.5}$$

$$y^i_{u,v} \in 0, 1 \tag{5.6}$$

Equation 5.1 is the objective function of the ILP. This equation represents the the summation of all weights of edges whose vertices are not within the same partition. Equation 5.2 ensures that each vertex is assinged to one partition. Equation 5.3 is used to determine whether vertex u and vertex v are in different partitions. Equation 5.4 ensures that all partitions are full, therefore not allowing vertices to exist outside partitions. Equations 5.5 and 5.6 are the constraints ensuring that x_u^i and $y_{u,v}^i$ are binary values.

We can then take these equations and transform them to represent an identical but further constrained problem of maximum K uncut total weight from the edges. This problem can be represented through the Equations 5.7, 5.8, 5.9, 5.10, 5.11 and 5.12. Where the only new parameter is $|E|$ indicating the total value of the set of edges within the graph. It should be noted that $|E|$ double counts edges. As it counts each edge from vertex u to vertex v as well as one from v to u.

$$minimize\ |E| - \sum_i \sum_{u,v} \omega(u,v) y_{u,v}^i \tag{5.7}$$

$$\sum_i x_u^i = 1, \forall u \tag{5.8}$$

$$y_{u,v}^i = x_u^i * x_v^i, \forall u, v, i \tag{5.9}$$

$$\sum_u x_u^i = s_i, \forall i \tag{5.10}$$

$$x_u^i \in 0, 1 \tag{5.11}$$

$$y_{u,v}^i \in 0, 1 \tag{5.12}$$

Within this reformulation of the problem, only equation 5.9 has changed from its initial form in the min K cut problem, Equation 5.3. In this new form, Equation 5.9 indicates whether or not the vertices u and v are within the same partition, making it the opposite value of that in the Equation 5.3.

5.2 Algorithmic Approximation

The local search algorithm is divided into two components. The first component is a greedy method that builds the initial partition structures. The second component then takes these partitions and swaps vertices that reduce the cut value.

5.2.1 Partitioning Portion of the Algorithm

The partitioning algorithm is a greedy method that is shown in both algorithms 1 and 2. This algorithm will produce the initial partitioning that while feasible will be improved by the swapping portion.

The parameters and variables used are as follows:
G: the set of vertices in the graph that are not yet assigned to a partition
S: the sets that represent the k partitions
s_i: the capacity of partition i
$\omega(u,v)$: the weight of the edge connecting vertices u and v
N: sets of vertices in G that are adjacent to vertices in the partitions

Algorithm 2 Max Part

Require: N_i, S_i, G
1: Find $u \in G$ such that $\omega(u,v) \leq \omega(u',v)$ for any u', $v \in N_i$
2: $S_i = S_i \cup \{u\}$
3: $G = G \setminus S_i$
4: **for** $v' \in G$ **do**
5: **if** $\omega(u,v') > 0$ **then**
6: $N_i = N_i \cup v'$
7: **end if**
8: **end for**

Algorithm 3 Partitioning Algorithm

Require: G, S

```
for i = 1 to k do
    Randomly select a vertex u ϵ G
    S_i = {u}
    G = G \ S_i
    for v ϵ G do
        if ω(u,v) > 0 then
            N_i = N_i ∪ {v}
        end if
    end for
end for
while G ≠ 0 do
    for i = 1 to k do
        if ||S_i|| < s_i then
            Max Part(N_i, S_i, G)
        end if
    end for
end while
```

5.2.2 Swapping Portion of the Algorithm

Following the termination of the partitioning algorithm in the previous section which allows the swapping algorithm to start with an initial solution, that while still inferior is better than a random partitioning would expect to be. The swapping algorithm checks each pairing of vertices between the partition S_i and those in partition S_j to see if Equation 5.13 is satisfied when i≠j. If the equation is satisfied, the two vertices positions are swapped, thereby reducing the cut value. The only parameter not described above is W; the total connected weight of vertex u with the vertices within the partition S_i.

$$W(u, S_i) + W(v, S_j) < W(u, S_j) + W(v, S_i) - 2\omega(u, v) \tag{5.13}$$

The swapping algorithm will terminate when i≠j and Equation 5.13 can no longer be satisfied. This occurs when every possible paring of vertices for the partitions S_i and S_j satisfies the equation 5.14 where i≠j. Note the parameters of Equation 5.14

represent the same information as those in 5.13.

$$W(u, S_i) + W(v, S_j) \geq W(u, S_j) + W(v, S_i) - 2\omega(u, v) \tag{5.14}$$

5.2.3 An Example

To further clarify the two portions of the algorithm, an example is provided in Figure 5.1.

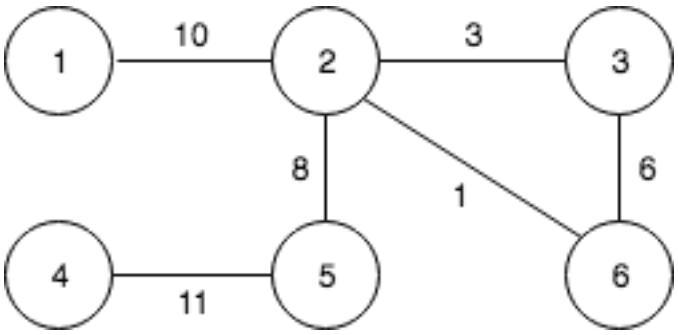

Figure 5.1: Example graph.

To begin, the algorithm requires the capacities of the partitions. For this example we will have two partitions of capacity 3, s_1=3 and s_2=3. We then proceed to choose random vertices for each partition. From Figure 5.1, we will choose vertex 1 for partition 1 and vertex 3 for partition 2. We then add vertices to the partitions in a greedy manner, which of the unassigned vertices has the highest total sum of weight from edges that connect to other vertices in the partition. Vertices are added to the partitions in the following order. Vertex 2 to partition 1. Vertex 6 to partition 2. Vertex 5 to partition 1. Lastly vertex 4 to partition 2, because there are no longer any connected unassigned vertices. Thus terminating the partitioning algorithm.

At this point the swapping algorithm will commence. At the beginning partition 1 will consist of the vertices (1,2,5) and partition 2 (3,6,4). The first and only swap will occur between vertex 1 and 4 making partition 1 become (4, 2, 5) and 2 (3, 6, 1). This results in the true minimum cut value of 14.

5.2.4 Paralellization

When considering paralellization of the above algorithm there are two avenues that we can explore. The first is the standard CPU based paralellization and the second

is the newer GPU based paralellization. CPU based paralellization has the ability of processing multiple threads of information at once that is capped by the number of cores in a given processors and whether they are capable of hyper threading. In GPU based parallelization the ceiling is much higher and therefore the potential speed gains are higher as well. The GPU is designed as a graphics tool and is therefore designed to have a high throughput to keep pixels updated on screen. Due to the required performance for graphics it can be useful in speeding up many algorithms. The ceiling of the GPU is not limited to the number of cores like a CPU but the number of blocks used, each of which has the capability of processing 32 threads at once. The main barrier that needs to be overcome when using GPU paralellization is the expensive kernel call that is needed utilize the GPU.

CPU Based

As the algorithm is divided between initial partitioning and swapping, their paralellizations will be discussed separately. The initial partitioning algorithm is already quite fast but there may be benefits in it's paralellization, and is therefore at least worth investigating. Conversely, the swapping portion of the algorithm is where the algorithm spends most of it's time and is the most likely to yield benefits of paralellization.

The paralellized version of the initial partitioning contains two changes from the sequential greedy version described previously. The first is the determining of adjacent nodes to the partition and the second is the determining the max part. The parrallelized algorithm for determining adjacent nodes can be seen in Algorithm 4. The parallelized algorithm for the max part can be seen in Algorithm 5.

Algorithm 4 Parallelized version for finding nodes adjacent to the partition that are unallocated.

Require: G, S_k

1: $adjacentBinary$ = array of 0's the size of G
2: create threads
3: id = thread number
4: $target = id$
5: **while** $target$ <size of partition S_k **do**
6: **for** $u \epsilon G$ **do**
7: **if** $\omega(S_k(target),u)$ **then**
 adj[u] = 1
8: **end if**
9: **end for**
10: $target$ += number of threads
11: **end while**
12: wait for threads to finish and continue sequentially
13: N = empty list
14: **for** $u \epsilon G$ **do**
15: **if** $adjacentBinary(u) == 1$ **then**
16: add u to N
17: **end if**
18: **end for**

Algorithm 5 Parallelized version of the max part algorithm in algorithm 2

Require: S_k, N

1: $bestUs$:list of 0's the length of the number of threads
2: $bestVs$:list of 0's the length of the number of threads
3: start threads
4: id = thread number
5: $target = id$
6: **if** id ¡ length of S_k length of N **then**
7: $bestUs(id) = id$ / length of N
8: $bestVs(id) = id$ % length of N
9: **end if**
10: **while** $target$ ¡ length of S_k length of $adjacent$ **do**
11: $newU = target$ / length of N
12: $newV = target$ % length of N
13: **if** $\omega(S_k(bestUs(id)),N(bestVs(id))) < \omega(S_k(bestUs(newU)),N(bestVs(newV)))$ **then**
14: $bestUs(id) = newU$
15: $bestVs(id) = newV$
16: **end if**
17: $target$ += number of threads
18: **end while**
19: wait for threads to finish
20: $topU = bestUs(0)$
21: $topV = bestVs(0)$
22: **for** i from 1 to length of $bestUs$ **do**
23: **if** $\omega(S_k(topU),N(topV)) < \omega(S_k(bestUs(i)),N(bestVs(i)))$ **then**
24: $topU = bestUs(i)$
25: $topV = bestVs(i)$
26: **end if**
27: **end for**

The parallelized part of the swapping portion can be seen in Algorithm 6. This represents the calculation of the four total connected weights, W, in 5.13. Therefore, this algorithm is used four times inside the necessary loops needed to check for the potential swaps between nodes and partitions. This algorithm essentially divides the

work of summation between multiple threads. So if there are 4 threads then each does a quarter of the sum. Once this is finished the threads each add their sums to a shared total with synchronization to ensure there are no race conditions.

Algorithm 6 CPU parallelized summation

1: $localSum = 0$
2: id = thread number
3: $target = id$
4: **while** $target < s_i$ **do**
5: $\quad index1 = S_k(u)$
6: $\quad index2 = S_k(target)$
7: $\quad localSum \mathrel{+}= \omega(index1, index2)$
8: $\quad target \mathrel{+}=$ number of threads
9: **end while**

GPU Based

Due to the expensive cost of calling the kernel for the GPU, the time taken to move data to the GPU and back, its more feasible to use it for larger cases. Since it will only be useful with larger cases, almost the entirety of the run time will be consumed by the swapping portion. Therefore, only a GPU based version of the swapping portion will be investigated.

The algorithm for the GPU based version of the swapping algorithm can be seen in Algorithm 7. This algorithm is essentially the same as Agorithm 6, however, was adapted slightly for the GPU. Unfortunately, all threads need to communicate with each other to properly maintain the partitions, therefore the full power of the GPU can not be harnessed as only one block can be used. The GPU does allow the execution of more threads simultaneously , for this reason a binary reduction was used. The binary reduction adds the back half of the array to the mirrored positions in the front half. The f̈unctional lengthöf the array is then reduced accordingly. This process continues until the total sum is in the 0th index of the array.

Algorithm 7 Partitioning Algorithm

Require: shared block variables array of integers *sums*, integer *sumsLength*,

```
1: sumLength = length of sums
2: id = thread number in the block
3: while sumLength >1 do
4:    target = id
5:    while target <sumLength do
6:       if target + 1 <sumLen then
7:          sums(target) += sums(sumLenght - target - 1)
8:       end if
9:       target += number of threads
10:   end while
11:   wait for all threads in the block to get to this point
12:   if id is 0 then
13:      sumLength = sumLength / 2 + sumLength % 2
14:   end if
15:   wait for all threads in the block to get to this point
16: end while
17: wait for all threads in the block to get to this point
```

5.3 Discussion

In this chapter the mincut problem was defined. An ILP model was established that was used to judge the quality of the algorithmic version. The algorithmic solution presented consists of an initial greedy partitioning followed by a swapping algorithm. This algorithm was presented as both a sequential version and a paralellized version. The later having both a CPU based version and a GPU based one.

The purpose of the proposed algorithm is to provide a time efficient approximation that could be used in a variety of clustering situations within the ED. In the following chapter the specifics of the algorithms performance will be discussed. The reason for the focus on time efficiency is to utilize it the algorithm as a tool to aid physicians in making decisions about patients through the areas discussed in the related works chapter in real time.

Chapter 6

Experimental Results

6.1 Physician Scheduling

6.1.1 Candidate Schedule Generation

The reality of schedule testing is that the problem space is extremely large. There are 9 shift start times that need to be allocated to a starting time and if shifts begin on a 15 minute interval (i.e., 0:00,0:15,0:30,0:45,1:00,...) there 96 possible start times for each physician. Then using the choose function it can be seen that the total possible combinations is on the order of 10^{11}. This is not a space that can be searched completely in a reasonable amount of time. Therefore, some restrictions are placed on the choosing of candidate schedules as some are likely to be not very useful, meaning that they do not properly meet demand, and some are impractical. Several logical

constraints can be imposed, while some are departmental scheduling rules to follow. The first simple logical constraint for the schedule is that a physician is required to be scheduled at any given point of the day. The reasoning for this constraint is clear and obvious. The next constraints that were proposed were based on physician input about realistic scheduling constraints for the ED. Firstly, it was decided that only one physician would start at any particular start time in the day. Secondly, that shifts would begin on half hour intervals and that there would be spacing of at least one hour between shift starting times. Thirdly, that the over night physicians start time (i.e., 0:00/24:00-7:00), would remain unchanged. These constrains greatly reduce the number of candidate schedules, however, not all of the remaining are worth considering. A few additional constraints can be applied to reduce the solution space that better defines an optimal schedule based on the different levels of patients through out the day, in order to avoid testing schedules that allocated large numbers of physicians to times with lower patient volumes.

This leaves approximately 454,000 possible schedules to examine. Since our goal is to determine the optimal schedule for the ED, allowing separate schedules for the weekdays and the weekend was important. Therefore, fully testing the combinations of all schedules is again too large. In order to remedy this the schedules were tested separately to determine the best performing schedules for weekdays and the weekends, the top 100. As we are considering two metrics for the schedules, PIA and LOS, a score is needed that considers both, the equation used can be seen in equation 6.1.

$$metric = \frac{the\ number\ of\ patients\ with\ PIA\ under\ 2\ hours}{number\ of\ patients} * \frac{the\ number\ of\ patients\ with\ LOS\ under\ 7\ hours}{number\ of\ patients} \tag{6.1}$$

6.1.2 Optimal Schedule

From the 10,000 schedules that consider both weekdays and weekends the resulting schedule that was found to be the optimal was:

For the Weekends:

- 0:00/24:00-7:00
- 5:30-13:30
- 7:30-15:30

- 10:00-18:00
- 12:00-20:00
- 14:00-22:00
- 16:00-24:00/0:00
- 18:00-2:00
- 20:30-4:30

For the Weekdays:

- 0:00/24:00-7:00
- 5:30-13:30
- 7:30-15:30
- 10:00-18:00
- 12:00-20:00
- 14:00-22:00
- 16:00-24:00/0:00
- 18:00-2:00
- 20:00-4:00

The difference between the two schedules is a half hour start time adjustment on the last shift.

Further investigation of the top 100 schedules showed that the shift starts that appear most often. The shifts for the weekday schedules can be seen in Figure 6.1. While the shifts for the weekend schedules can be seen in Figure 6.2. The most frequently occurring start times for both weekdays and weekends are 10:00, 12:00, 14:00 as they are in nearly every one of these top 100 schedules. It should be noted that all of these top 100 schedules had a metric rating over 0.96 according to equation 6.1.

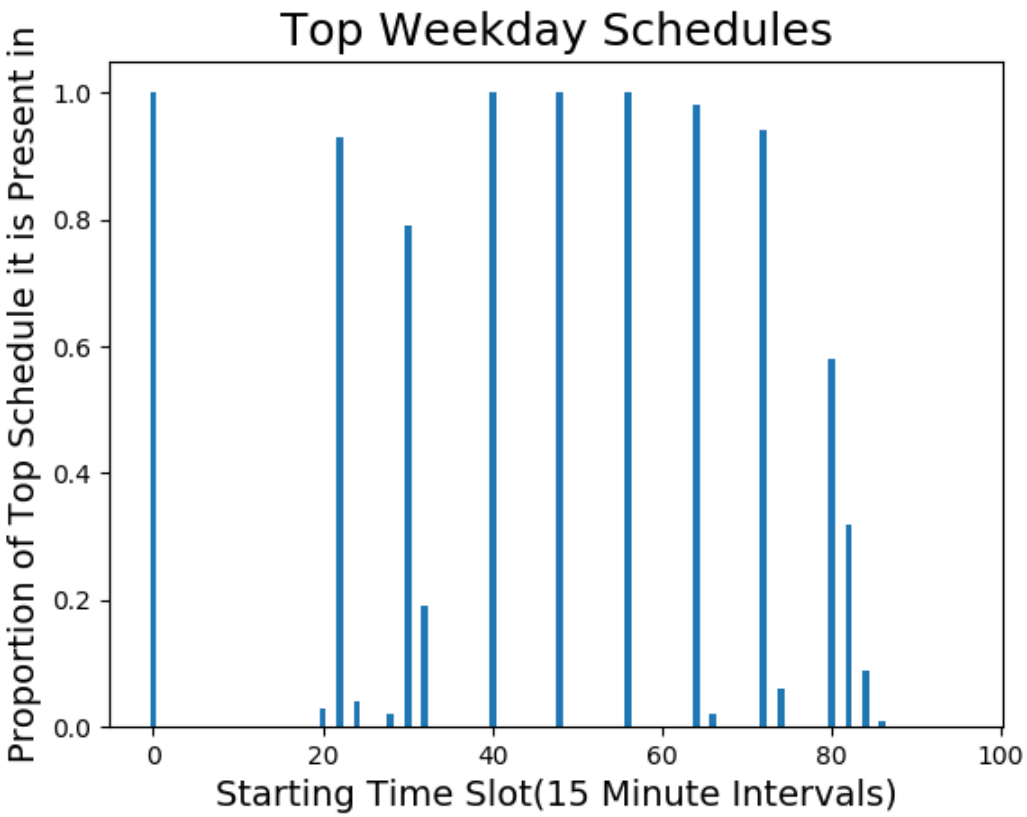

Figure 6.1: Proportion of the top 100 weekday schedules that occur most frequently.

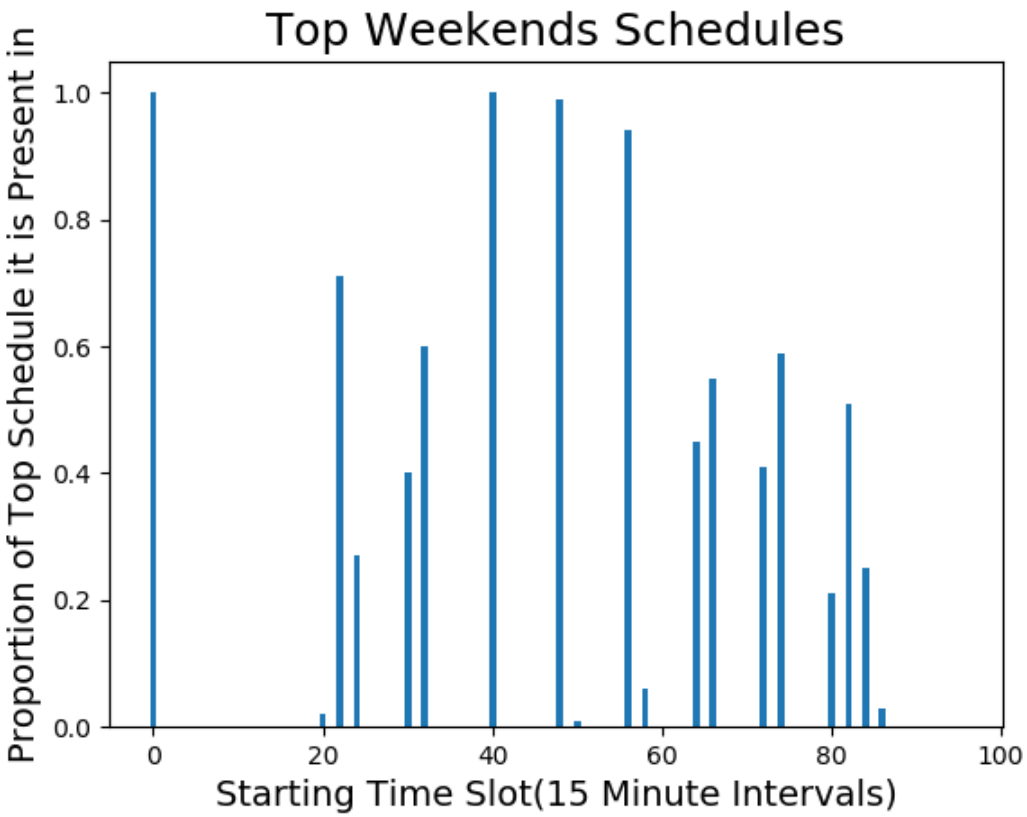

Figure 6.2: Proportion of the top 100 weekend schedules that occur most frequently.

A network graph was generated to show the relationship among the top schedules and how they interact with each other. The graph representing weekday schedules can be seen in Figure 6.3. The graph representing weekend schedules can be seen in Figure 6.4. It can be seen that in the case of weekdays, the start times; 5:30, 7:30, 10:00, 12:00, 14:00, 16:00, 18:00 and 20:00 are all highly correlated. While in the case of weekends start times; 5:30, 8:00, 10:00, 12:00, 14:00, 16:30, 18:30 and 20:30 are highly correlated.

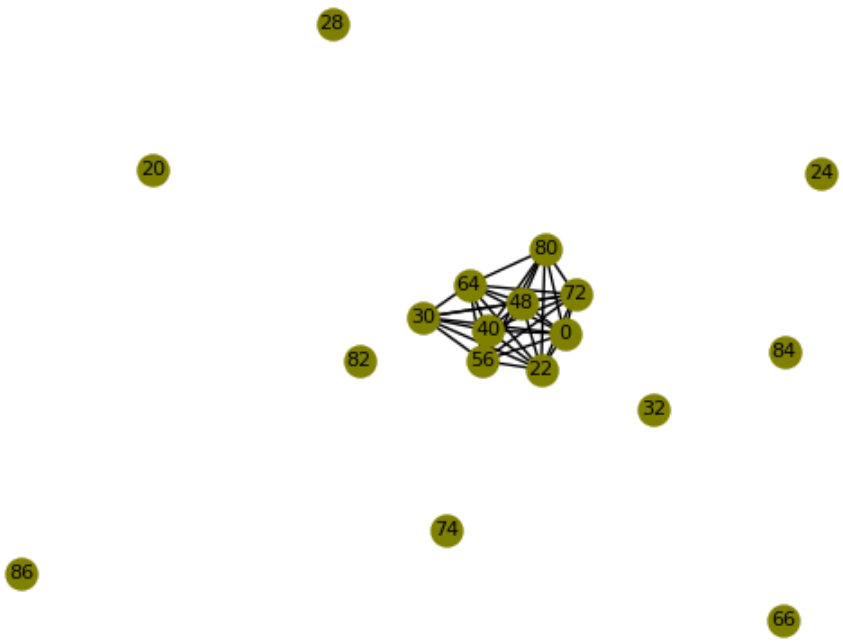

Figure 6.3: A graph representing shifts that appear together in at least 50% of the top 100 weekday schedules.

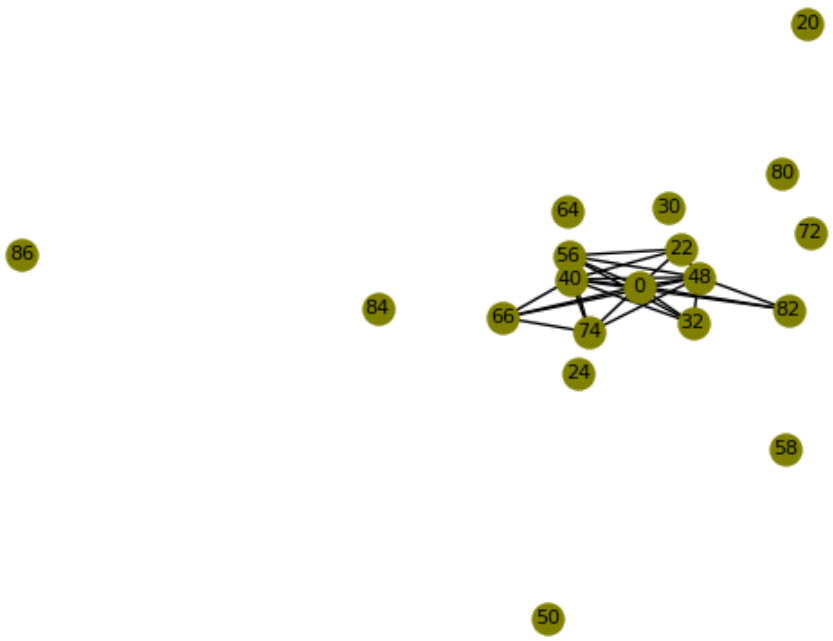

Figure 6.4: A graph representing shifts that appear together in at least 50% of the top 100 weekend schedules.

6.2 Cluster Partitioning

In this section the results of testing the mincut algorithm will be discussed. The algorithm was first compared to an ILP solution in order to determine how well it is able to approximate the mincut of a graph. Second the investigation turns to the parallelization of the algorithm.

6.2.1 Graph Generation

In order to test the algorithm, graphs are required. This study used undirected graphs and tested the algorithm on both weighted and unweighted graphs. The process of the graph generation can be seen in Equation 8. The first of these loops forms a graph in which every node can reach another node through some path of edges. The two nested loops add additional edges to make the minimum cut harder to solve.

Algorithm 8 Partitioning Algorithm

Require: V

```
for i ϵ V do
    select a random node less than i
    form an edge with it
end for
for i ϵ V do
    for j ϵ V do
        if no edge is present between i and j and i != j then
            Give a 20% chance to add an edge between i and j
        end if
    end for
end for
```

The unweighted graphs were simply created by setting all edge values in the weighted graphs to one.

6.2.2 Algorithmic Performance

First the results of the algorithm were compared to those of the ILP model. Scenarios with graphs of sizes 20, 30, 40, 50, 80 and 100 nodes were tested. These graphs where partitioned into sets of 2, 3, 4 and 5 partitions. This was done for both weighted and unweighted versions of the graphs. There were 3 sets of each of the graph size and weighted/unweighted combinations used to avoid testing on a graph that was easy for the algorithm to solve. These tests were all performed 20 times to find a mean for the minimum cut from the algorithm, each time using a random seed for the initial partitioning. The ILP tests were each run once for a maximum period of 3 hours.

In Figure 6.5 the comparison between the increase in graph size and the mincut values of both the algorithm and ILP for weighted graphs can be seen. The ILP problem was solved using the Gurobi solver and found two curves, the incumbent and best bound. These are the two values that the solver uses to constrain the problem when solving it. The incumbent is the value of the best solution found so far. While the best bound is what the solver believes to be the optimal value at this point in the execution. When these two curves share values the solver has reached a solution which it believes to be optimal. The algorithm curve follows the incumbent

curve very closely in all three scenarios and then deviates when the number of nodes in the graph reaches 50. This can be seen in the graph in the bottom right of the figure where the 95% confidence interval of the mean is very close to 1.00, the value of the incumbent, except in the case of the 20 node graph. When the case of the 20 node graph is examined in the other three graphs, it can be seen that they are very close. The proportion maybe misleading as it could easily be the result of a couple of nodes out of place. The best bound curve becomes flat when the number of nodes in the graph becomes large. This is due to how the effort required by Gurobi to bring down the best bound compared with finding a better incumbent. When the logs for Gurobi where examined it was found that the solver often found it's best incumbent, the solution it reported in the end, and then spent a significant portion of time bringing down the best bound. Therefore the incumbents are most likely far more closer to the optimal then they appear.

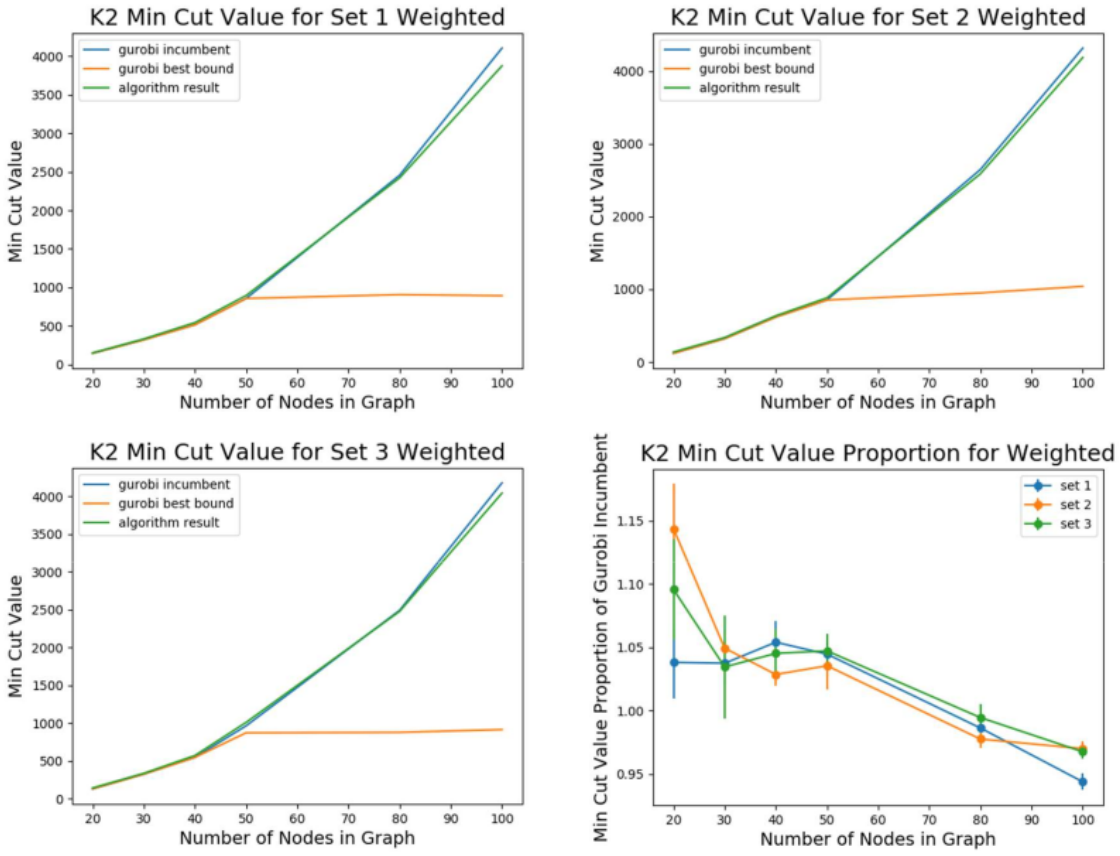

Figure 6.5: A graph showing the relationship between algorithmic performance in finding the optimal solution and the number of nodes in the weighted graphs.

Examining Figure 6.6 we can see that the algorithm also performs well with the scaling of the number of partitions. Note the result shown in the first data points in each line in the previous figures are the same as those in these. Additionally these graphs only have one curve due to Gurobi solving to what it believed to be optimal for all the test shown.

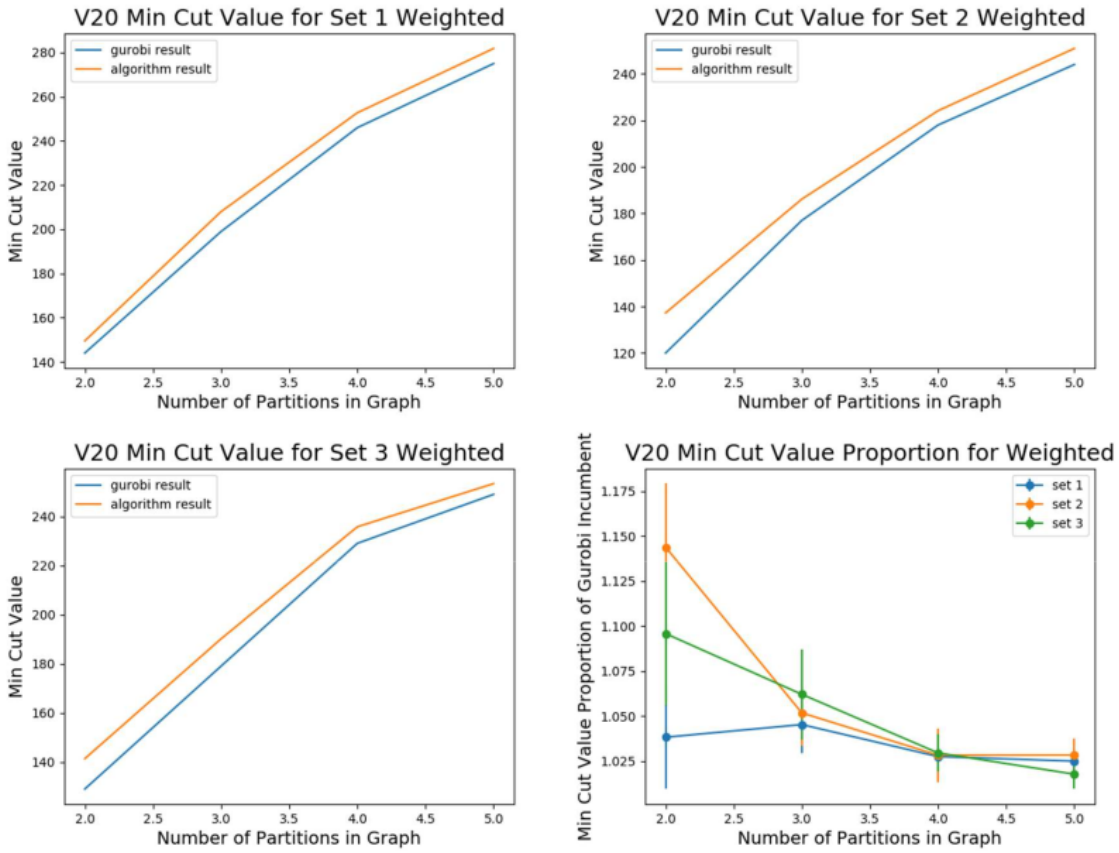

Figure 6.6: A graph showing the relationship between algorithmic performance in finding the optimal solution and the number of partitions in the weighted graphs.

The results for the unweighted graphs can be seen in Figures 6.7 and 6.8. The algorithm performed well compared to the incumbent with respect to graph size. It did, however, have more difficulty with respect to the number of partitions in the graph.

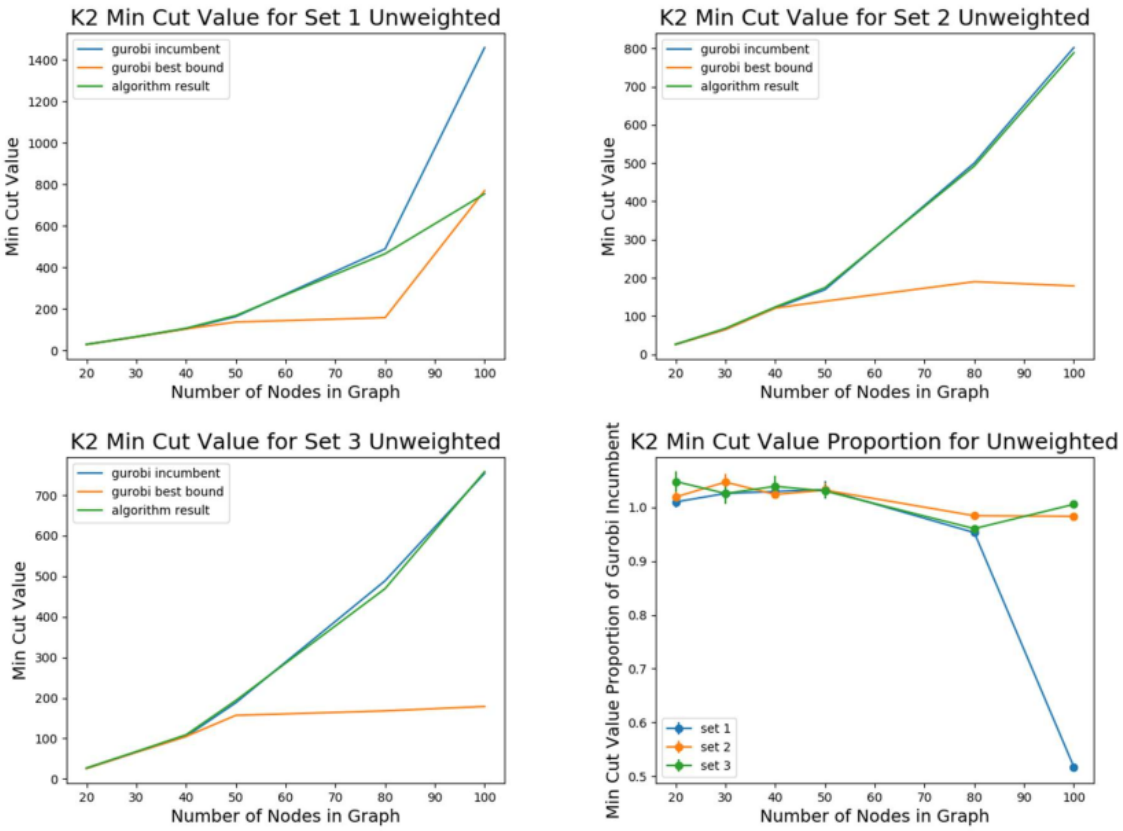

Figure 6.7: A graph showing the relationship between algorithmic performance in finding the optimal solution and the number of nodes in the unweighted graphs

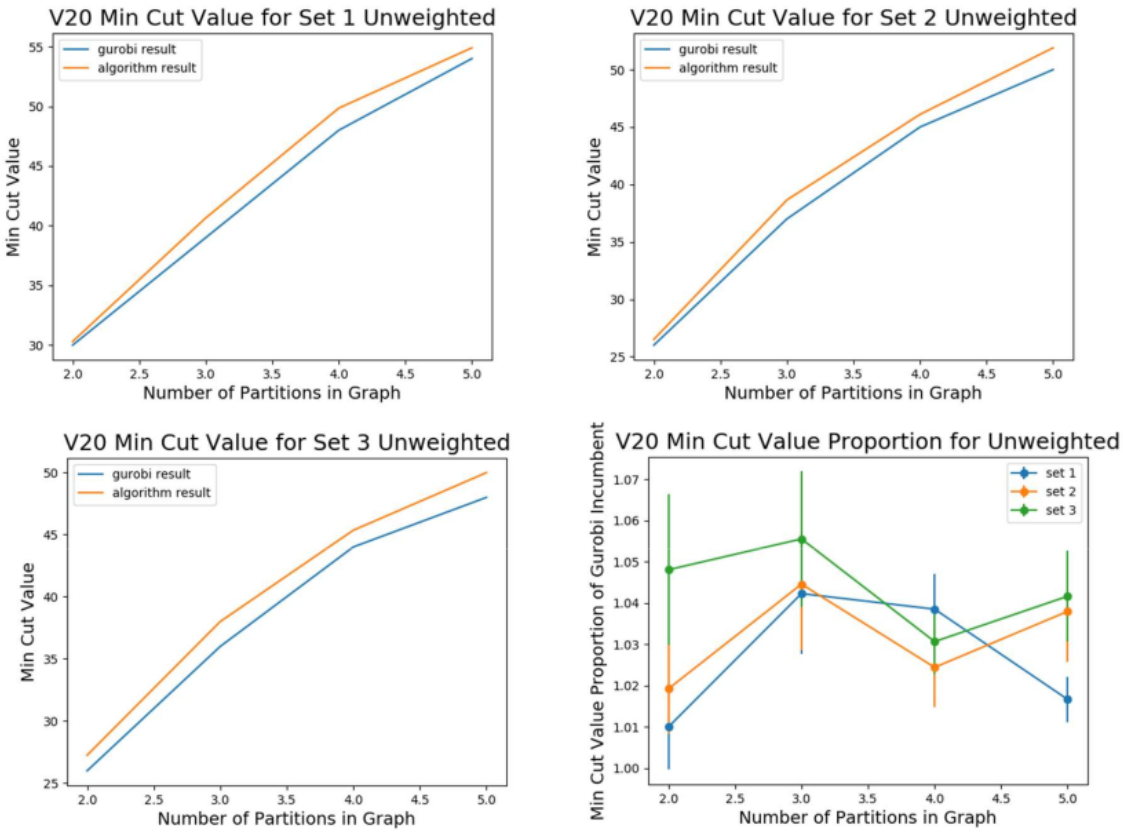

Figure 6.8: A graph showing the relationship between algorithmic performance in finding the optimal solution and the number of partitions in the unweighted graphs.

All the data points shown in Figures 6.5 and 6.6 can be seen in Table 6.1 below.

Table 6.1: A table showing the Algorithmic and ILP performance comparisons for finding the optimal.

Set	K	V	Gurobi Iccum-bent	Gurobi Best Bound	Gap	Algorithm Cut Value	C.I. 95%
1 weighted	2	20	144	144	0%	149.5	145.31-153.59
1 weighted	2	30	319	319	0%	330.95	325.48-336.41
1 weighted	2	40	515	515	0%	542.85	534.34-551.36

1 weighted	2	50	856	856	0%	894.25	886.06-902.44
1 weighted	2	80	2456	906	63.11%	2422.4	2400.85-2443.95
1 weighted	2	100	4106	892	78.28%	3875.45	3848.78-3902.12
1 weighted	3	20	199	199	0%	208.0	204.82-211.18
1 weighted	4	20	246	246	0%	252.75	249.846-255.65
1 weighted	5	20	275	275	0%	281.85	279.60-284.10
2 weighted	2	20	120	120	0%	137.25	133.00-141.50
2 weighted	2	30	321	321	0%	336.85	331.59-342.11
2 weighted	2	40	618	618	0%	635.70	630.13-641.27
2 weighted	2	50	852	852	0%	882.25	866.52-897.97
2 weighted	2	80	2647	950	64.11%	2587.75	2569.98-2605.52
2 weighted	2	100	4315	1040	75.90%	4185.75	4159.54-4211.96
2 weighted	3	20	177	177	0%	186.15	182.92-189.38
2 weighted	4	20	218	218	0%	224.15	220.88-227.42
2 weighted	5	20	244	244	0%	250.90	248.63-253.17
3 weighted	2	20	129	129	0%	141.35	136.25-146.45

3 weighted	2	30	324	324	0%	335.20	322.07-348.33
3 weighted	2	40	546	546	0%	570.7	559.86-581.54
3 weighted	2	50	963	871	9.55%	1008.5	995.57-1021.43
3 weighted	2	80	2490	876	64.82%	2476.45	2450.02-2502.88
3 weighted	2	100	4178	913	78.15%	4042.50	4019.28-4065.72
3 weighted	3	20	179	179	0%	190.10	185.64-194.56
3 weighted	4	20	229	229	0%	235.75	233.35-238.15
3 weighted	5	20	249	249	0%	253.4	251.45-255.35

The solution speed resulting from the algorithm rather than the ILP was assessed. In Figure 6.9 the runtimes and the number of nodes in the graph were compared. In the first graph, the runtime for the algorithm appears constant but when looking at the second graph it can be seen that they are not constant but instead grow slowly in comparison to the ILP ones. Note that when the ILP curves flatten out in higher cases it is due to the 3 hour time limit being reached. This occurs in both Figure 6.9 and Figure 6.11.

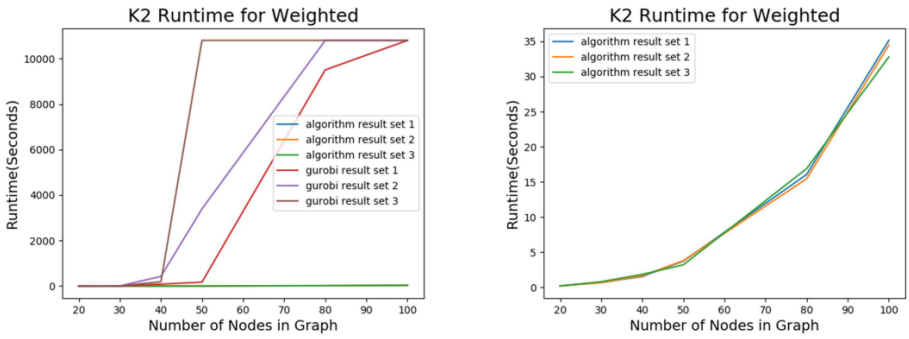

Figure 6.9: A graph showing the relationship between algorithmic performance in runtime and the number of nodes in the weighted graphs.

Figure 6.10 shows the corresponding results for the comparison of the number of partitions in the graph. The comparison between ILP and algorithm curves is similar. However, in this case the algorithm curves appear logarithmic rather than exponential, but most likely linear or a less severe exponential. This result is not unexpected given that there are more iteration loops to go through to check that there are no more partition combinations that must be considered but less node combinations in the this case.

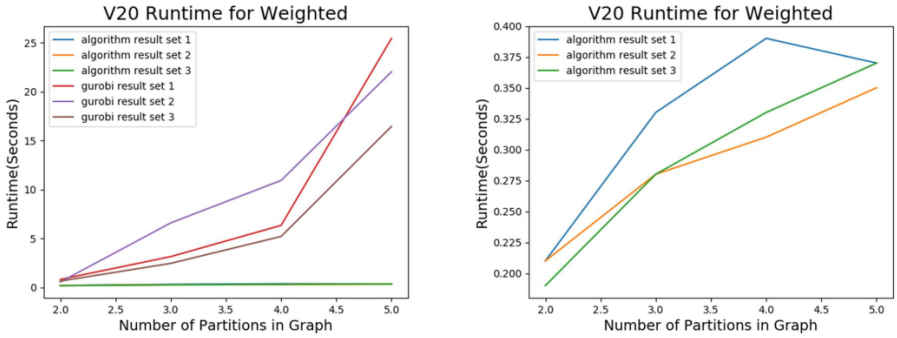

Figure 6.10: A graph showing the relationship between algorithmic performance in runtime and the number of partitions in the weighted graphs.

Very similar results can be seen if Figures 6.11 and 6.12 for the unweighted graphs.

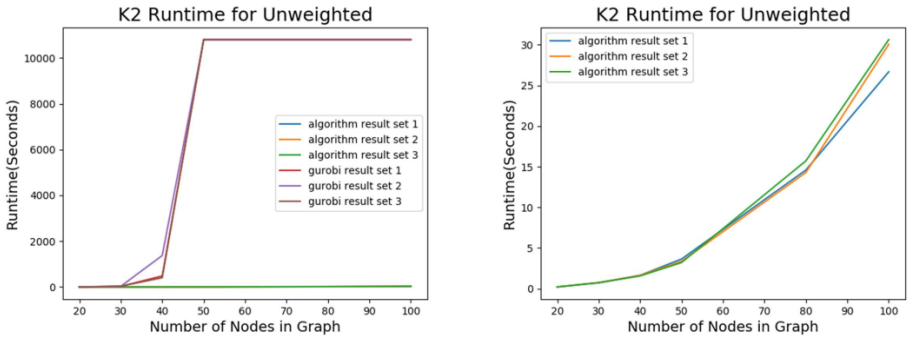

Figure 6.11: A graph showing the relationship between algorithmic performance in runtime and the number of nodes in the weighted graphs.

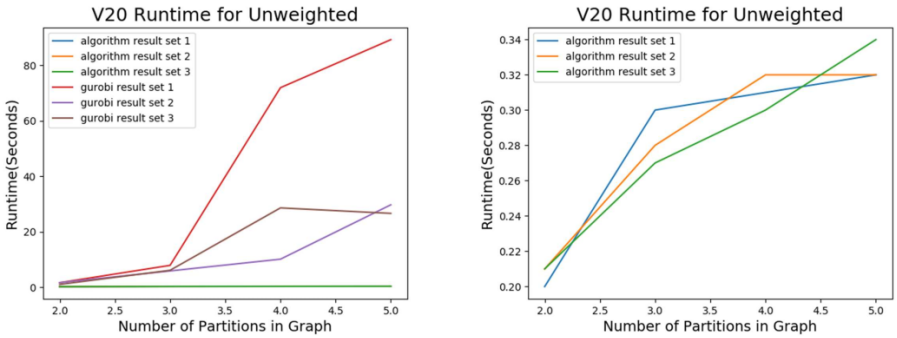

Figure 6.12: A graph showing relationship between algorithmic performance in runtime and the number of partitions in the weighted graphs.

All the data points shown in Figures 6.9 and 6.10 can be seen in table 6.2 bellow. Note Gurobi time indicates the runtime of the Gurobi solver.

Table 6.2: A table showing the Algorithmic and ILP performance comparisons for runtime.

Set	K	V	Gurobi Time(seconds)	Algorithm Time(seconds)
1 weighted	2	20	0.83	0.21
1 weighted	2	30	8.84	0.69
1 weighted	2	40	171.70	1.54
1 weighted	2	50	9504.54	3.73
1 weighted	2	80	10800.15	16.05
1 weighted	2	100	10800.09	35.16

1 weighted	3	20	3.16	0.33
1 weighted	4	20	6.35	0.39
1 weighted	5	20	25.44	0.37
2 weighted	2	20	0.57	0.21
2 weighted	2	30	6.40	0.66
2 weighted	2	40	422.41	1.64
2 weighted	2	50	3392	3.80
2 weighted	2	80	10800.09	15.44
2 weighted	2	100	10800.15	34.49
2 weighted	3	20	6.60	0.28
2 weighted	4	20	10.94	0.31
2 weighted	5	20	22.03	0.35
3 weighted	2	20	0.66	0.19
3 weighted	2	30	4.74	0.80
3 weighted	2	40	191.78	1.85
3 weighted	2	50	10800.05	3.21
3 weighted	2	80	10800.15	16.88
3 weighted	2	100	10800.11	32.80
3 weighted	3	20	2.47	0.28
3 weighted	4	20	5.22	0.33
3 weighted	5	20	16.45	0.37

Overall, the results show that the algorithm could be used as an effective tool to quickly partition patients into groups. This could be used in real time to assist physicians in a variety of situations, as were mentioned in the related work section. The ILP model on the other would not be useful in a real time environment except for very small cases, as the runtime alone would render many results useless as decisions have already been made for the patients by that time.

6.2.3 Effects of Paralellization

As the algorithm is divided between the initial partitioning and the swapping component, the paralellization was divided between the two as well. Furthermore, since the

algorithm is dependant on a random seeding of each partition in the initial partitioning portion, then the test must be done in a manner that this seeding is the same for each test for a given V and K combination. This means that the initial partitioning test use the same seed and that the swapping tests use the same initial partition starting point. The resulting test for each of the chosen V and K combinations discussed below were generated by running each case 10 times. Some mean runtimes are displayed below in Table 6.3 for the initial partitioning portion and Table 6.4 for the swapping portion. These tests were all performed on a computer with the following specifications: 12 GB of RAM 2400MHz, an Intel i5-7300HQ CPU 2.50Ghz with 4 cores and a NVIDIA GeForce GTX 1050 GPU. Once again all implementations were done in C++. The GPU parallelization used CUDA and the CPU parallelization used openMP. Furthermore, due to limitations in RAM speed , the implementations use malloc memory blocks rather than objects to avoid using the heap. To ensure a consistent comparison, the sequential algorithm is also implemented this way. To further reduce computing overhead, the CPU based version utilized the same set of threads so new ones did not need to be created each time. Therefore, in the swapping portion the threads were all created prior to beginning to loop through nodes and partitions and syncronization is maintained between them all throughout. Similarly, the same was done with the GPU based swapping portion, note this only requires one call to the GPU to begin, minimizing it's expense. Additionally, in the CPU implementation each thread has it's own copy of information, such as the weight matrix. This was in response to having each thread using shared memory during testing, causing too many faults due to memory row refreshing. Also, it must be noted that these tests were run with no other programs active to provide the fairest comparison possible.

To begin a sequential algorithm, 2 thread, 4 thread and 8 thread case were tested for the initial partitioning portion. In Figures 6.13, 6.14, 6.15 and 6.16 the relationship between the number of nodes in the graph and runtime is presented. These are for K2, K3, K4 and K5, respectively. The plots demonstrate that as the gap between the methods narrows, the number of partitions increases. This is due to the fact that the parrellelization happens within the partitions as information is gathered about other nodes in the graph. Therefore, the gap does not shrink but merely lags behind in growth. Also, of note is that the 4 thread and 8 thread tests follow the same path. This shows that the process is a very busy one, with not much time to allow other threads to make use of the same core. In fact the speed of memory here is

most likely the limiting factor. Therefore, with better specifications this could result in additional speed. It must be considered though the initial partitioning process is already very fast so parallelization only becomes viable at very large scales or if it is a heavily repeated process. Furthermore the issue with in the case of larger case a higher percentage of the runtime is taken up by the swapping portion.

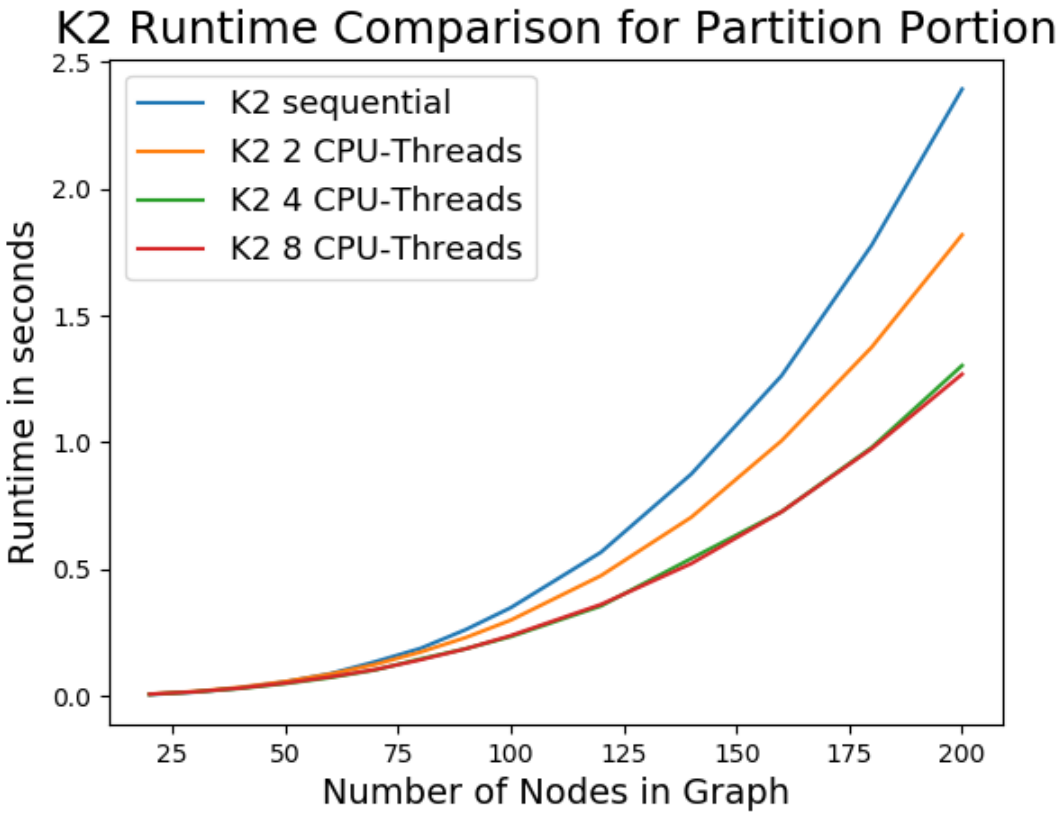

Figure 6.13: A graph showing the relationship between parallelization performance in runtime and the number of nodes in the K2 graphs in the initial partitioning portion.

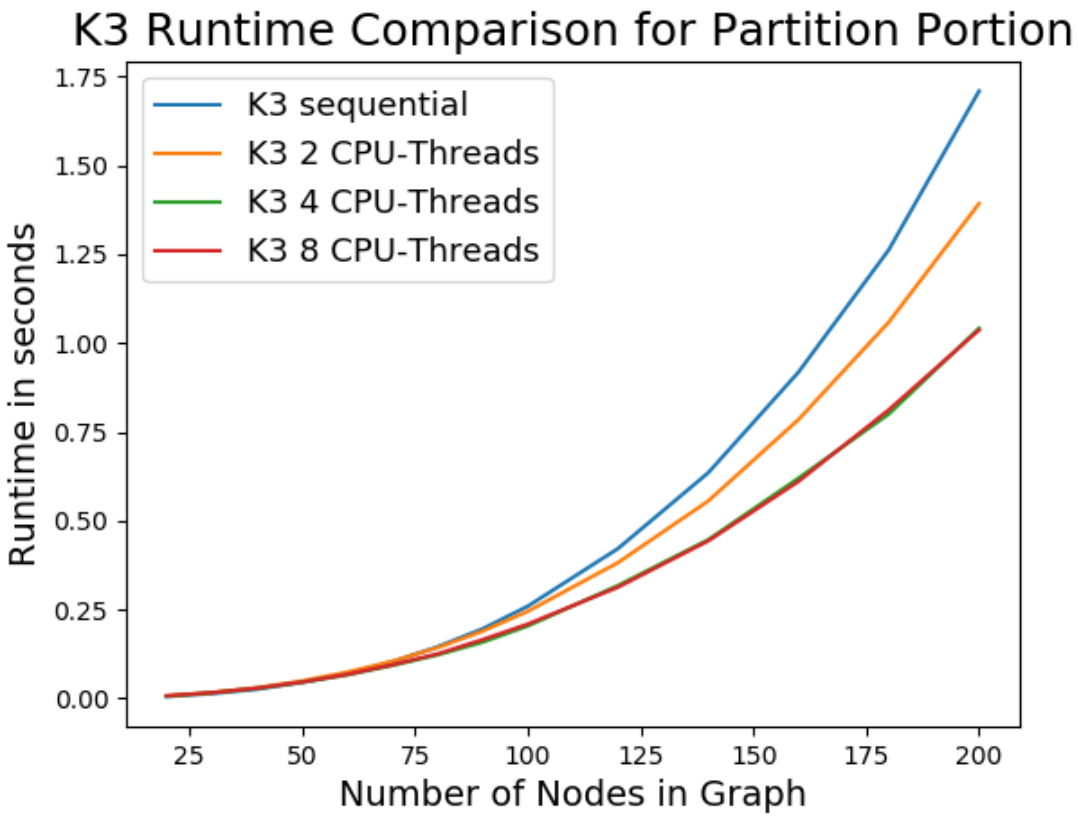

Figure 6.14: A graph showing the relationship between parallelization performance in runtime and the number of nodes in the K3 graphs in the initial partitioning portion.

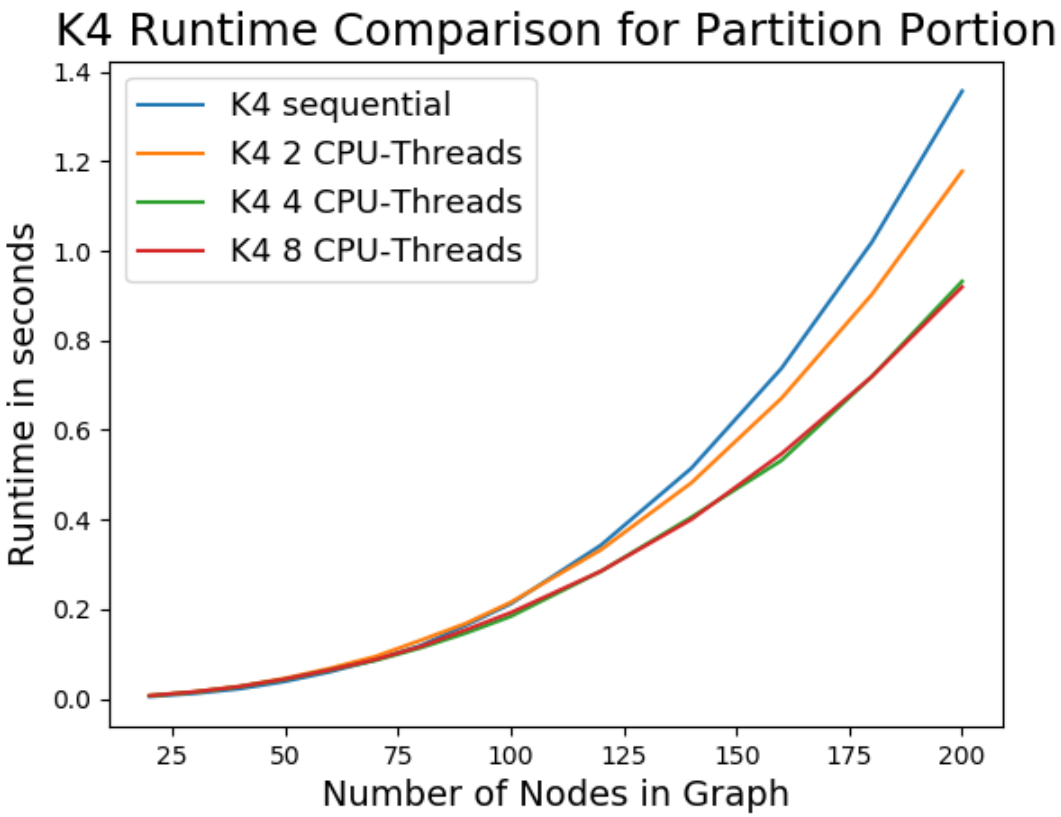

Figure 6.15: A graph showing the relationship between parallelization performance in runtime and the number of nodes in the K4 graphs in the initial partitioning portion.

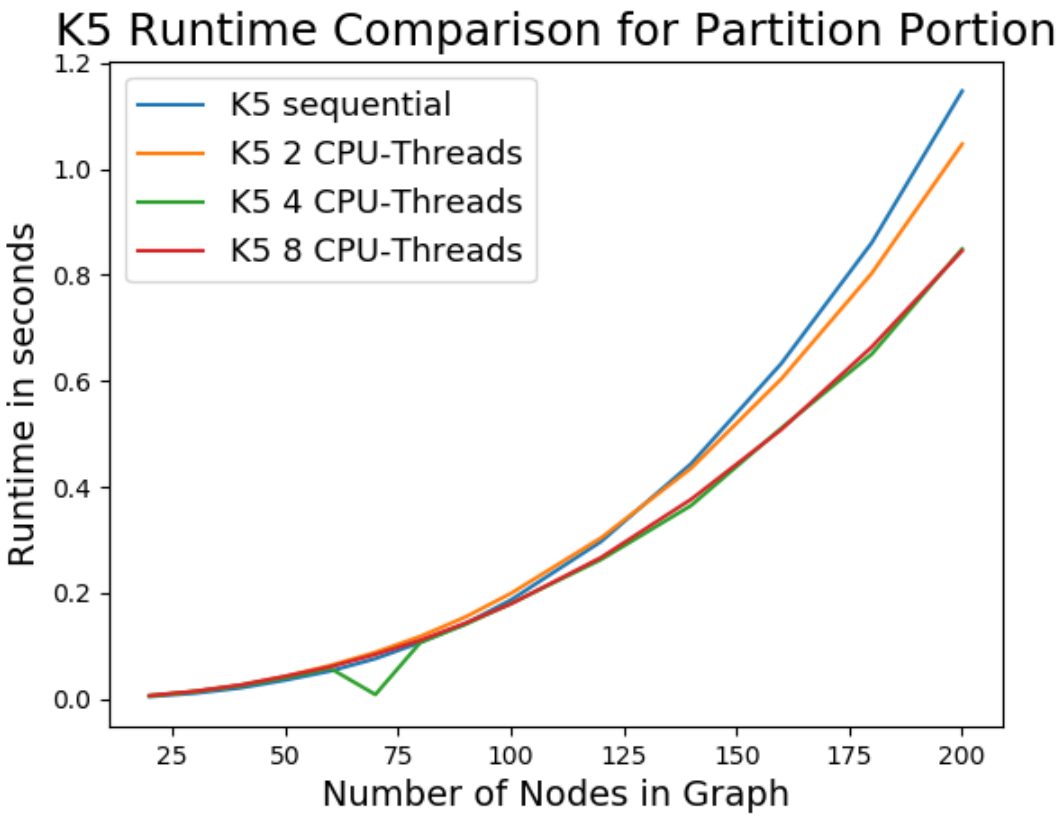

Figure 6.16: A graph showing the relationship between parallelization performance in runtime and the number of nodes in the K5 graphs in the initial partitioning portion.

Table 6.3: A table showing the comparison of runtimes for the initial partitioning portion of the problem.

K	V	Sequential Time (Seconds)	2 Threads Time (Seconds)	4 Threads Time (Seconds)	8 Threads Time (Seconds)
2	20	0.0064	0.008	0.0076	0.008
2	30	0.0163	0.0184	0.017	0.0174
2	40	0.0323	0.0348	0.0305	0.0319
2	50	0.0565	0.0577	0.0496	0.0517
2	60	0.0887	0.0872	0.0733	0.076
2	70	0.1359	0.1255	0.1035	0.1051

2	80	0.1884	0.1739	0.1467	0.1434
2	90	0.2622	0.2309	0.187	0.1865
2	100	0.3488	0.2996	0.2348	0.2387
2	120	0.5675	0.4751	0.3556	0.3615
2	140	0.8754	0.7052	0.5426	0.5225
2	160	1.2628	1.0066	0.7269	0.7264
2	180	1.7781	1.3757	0.9805	0.9758
2	200	2.3924	1.8181	1.303	1.269
3	20	0.0053	0.0072	0.0068	0.0076
3	30	0.0137	0.0162	0.0159	0.0163
3	40	0.0258	0.03	0.0283	0.029
3	50	0.0455	0.0493	0.0455	0.0462
3	60	0.0694	0.074	0.0662	0.0676
3	70	0.1034	0.1046	0.0929	0.0947
3	80	0.1451	0.1428	0.1225	0.1253
3	90	0.196	0.1894	0.1583	0.1655
3	100	0.2594	0.2451	0.2039	0.2091
3	120	0.4218	0.3822	0.3177	0.3132
3	140	0.6351	0.5557	0.4466	0.4434
3	160	0.9183	0.7848	0.6199	0.6109
3	180	1.2624	1.0588	0.8003	0.812
3	200	1.7089	1.3925	1.0415	1.0367

Next, the swapping portion of the algorithm with the same cases were considered as the initial partitioning portion with the additional GPU cases. Figures 6.17, 6.18, 6.19 and 6.20 show the relationship between the number of nodes in the graph and runtime. These are for K2, K3, K4 and K5 respectively. The GPU based solution can be quickly discounted as a viable form of parallelization, as it's scaling is very poor. While the GPU itself is very useful the cost of transferring the data is too great. In most cases this would be eventually overcome when larger cases are considered but the issue in this case is the growing weight matrix size with larger case causing the rapidly increasing trajectory of the GPU curves. Turning to the CPU based parallelizations runtime decreases along with the number of threads used, making

sequential the fastest. Therefore the overhead of keeping the threads synchronized with each other is too costly, showing that the parallelized swapping portion is not a viable option.

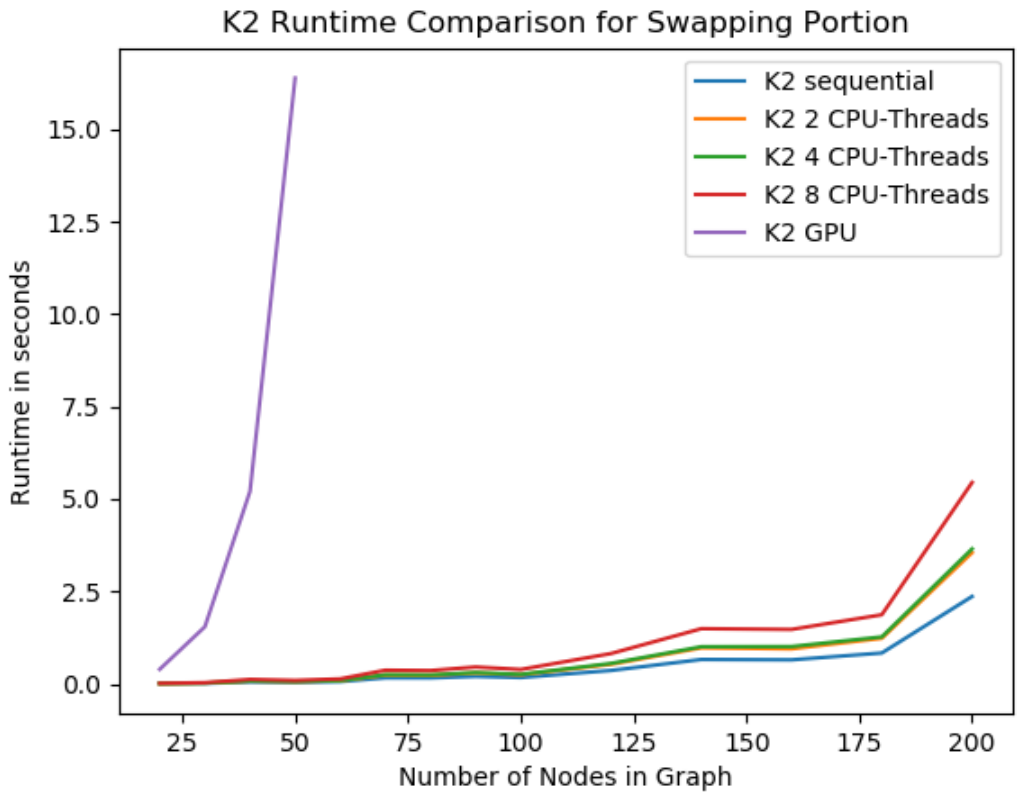

Figure 6.17: A graph showing the relationship between parallelization performance in runtime and the number of nodes in the K2 graphs in the swapping portion.

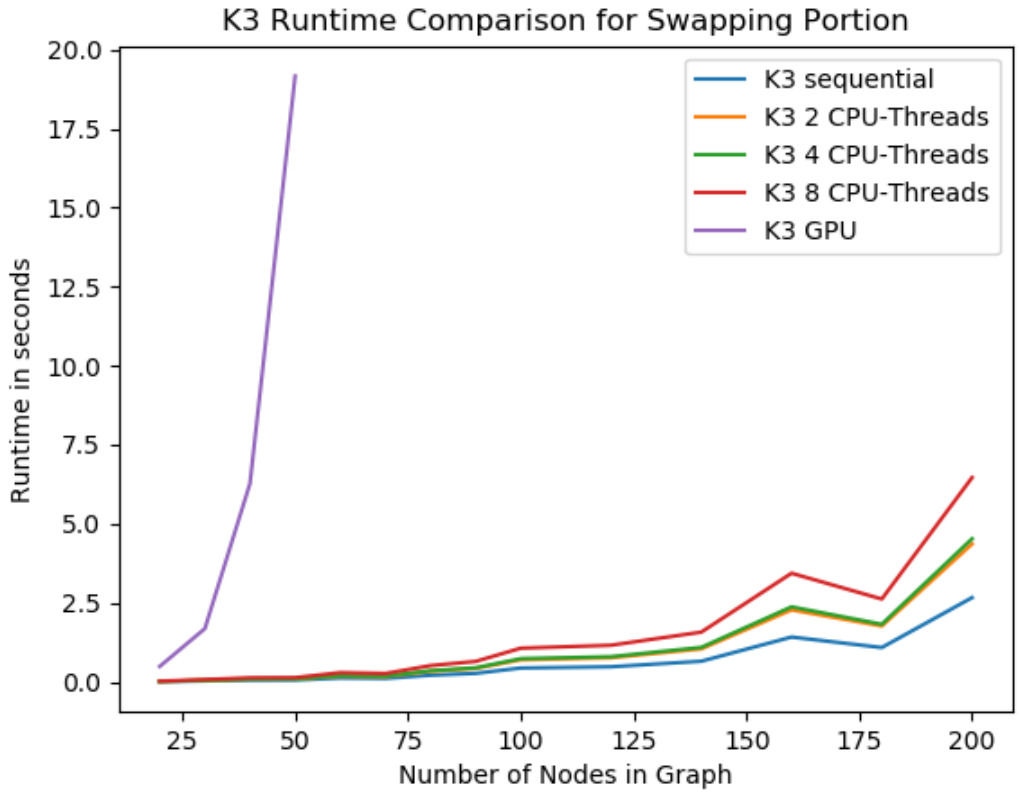

Figure 6.18: A graph showing the relationship between parallelization performance in runtime and the number of nodes in the K3 graphs in the swapping portion.

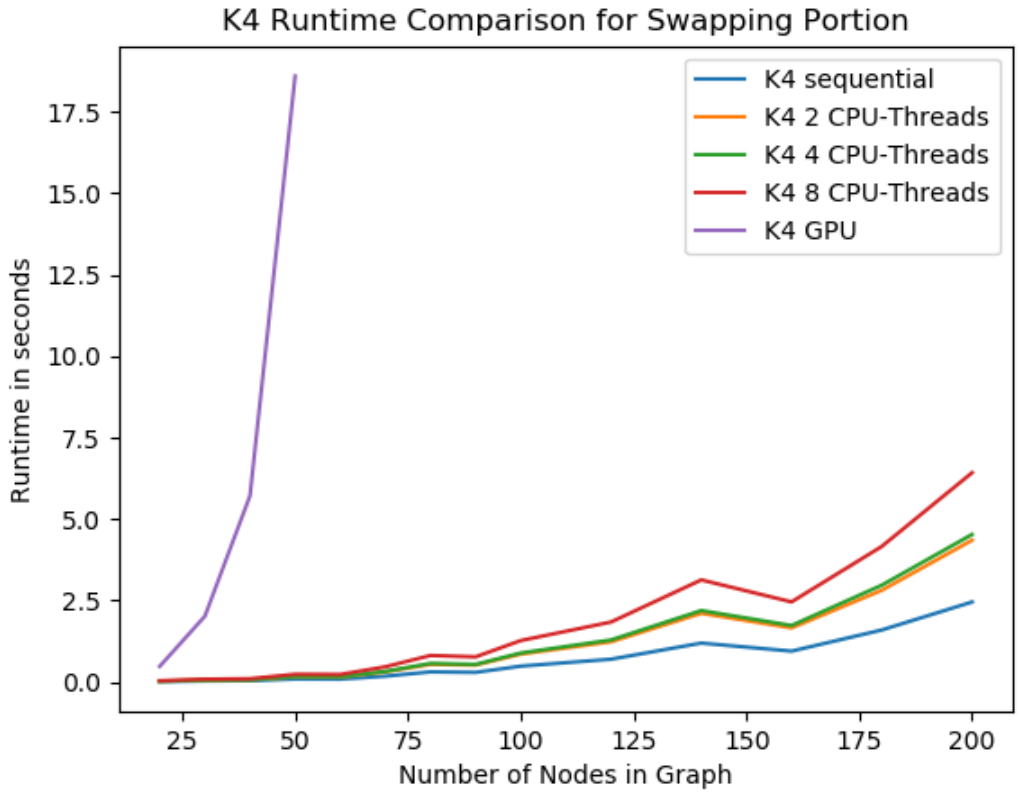

Figure 6.19: A graph showing the relationship between parallelization performance in runtime and the number of nodes in the K4 graphs in the swapping portion.

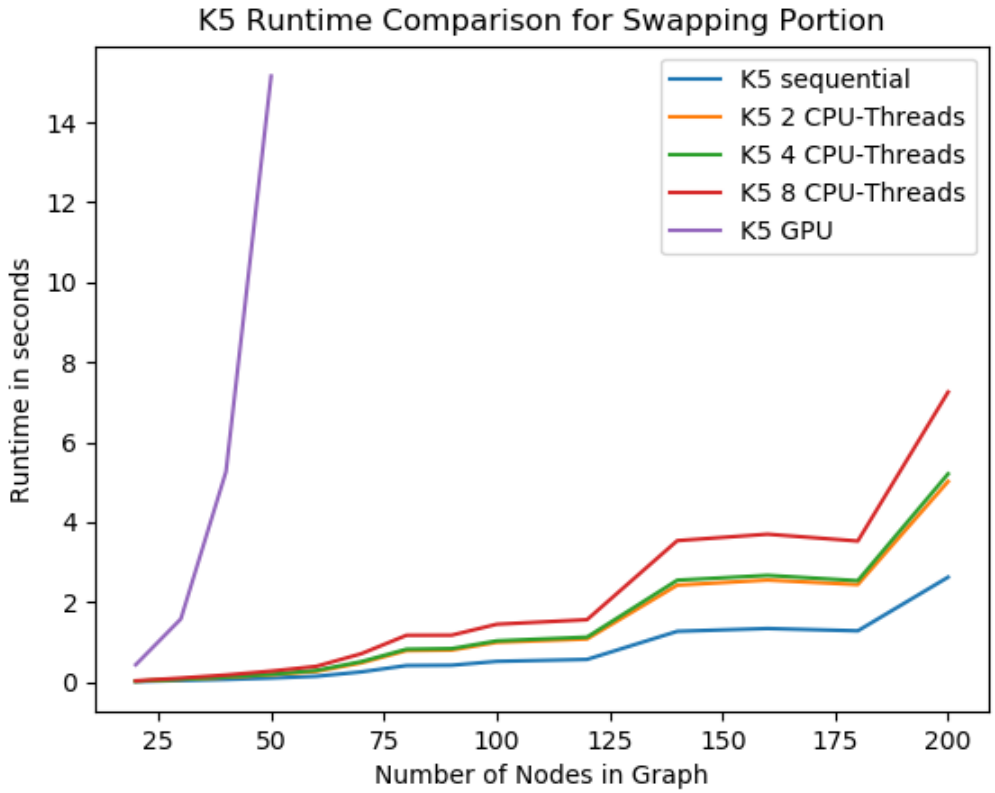

Figure 6.20: A graph showing the relationship between parallelization performance in runtime and the number of nodes in the K5 graphs in the swapping portion.

Table 6.4: A table showing the comparisons of runtime for the swapping portion.

K	V	Sequential Time (Seconds)	2 Threads Time (Seconds)	4 Threads Time (Seconds)	8 Threads Time (Seconds)	GPU Time (Seconds)
2	20	0.0178	0.0097	0.0105	0.0159	0.3966
2	30	0.016	0.022	0.0238	0.0347	1.5455
2	40	0.0543	0.0763	0.0793	0.1216	5.2066
2	50	0.0439	0.0601	0.0637	0.0955	16.3841
2	60	0.0631	0.0877	0.0916	0.1366	
2	70	0.1636	0.2383	0.2483	0.3717	
2	80	0.1623	0.2317	0.2422	0.3622	

2	90	0.2035	0.2963	0.3072	0.4577	
2	100	0.1729	0.2478	0.2581	0.3913	
2	120	0.3636	0.53	0.5548	0.8215	
2	140	0.6606	0.9752	1.0067	1.4939	
2	160	0.6512	0.9467	1.011	1.4741	
2	180	0.8364	1.2334	1.2726	1.871	
2	200	2.3652	3.5494	3.6512	5.4478	
3	20	0.0094	0.014	0.0148	0.0219	0.4884
3	30	0.031	0.0468	0.0491	0.0726	1.684
3	40	0.0543	0.0837	0.0877	0.1293	6.2748
3	50	0.0564	0.087	0.0929	0.1341	19.1629
3	60	0.1207	0.1863	0.196	0.2917	
3	70	0.109	0.171	0.181	0.2645	
3	80	0.2107	0.3333	0.3533	0.5145	
3	90	0.2688	0.4258	0.4442	0.647	
3	100	0.4381	0.6983	0.7341	1.0657	
3	120	0.4776	0.7622	0.7968	1.1587	
3	140	0.6526	1.0424	1.0901	1.5735	
3	160	1.4181	2.2777	2.3731	3.4366	
3	180	1.0852	1.7657	1.8215	2.6198	
3	200	2.6649	4.3638	4.5278	6.4635	

Overall, in my opinion the parallelization of the algorithm does not seem viable in practice as the only point that it is faster is during the initial partitioning. Which as mentioned becomes a smaller portion of the runtime as the number of nodes in the graph grows. A more useful of parallelization in my opinion would be to run several instances at once. This is because the use of different initial partitions could result in finding a better min cut value. Running these cases in parrallel would take roughly the same amount of time as running one sequentially as they require no synchronization between them. Additionally since they would all share a weight matrix the GPU may become viable. An issue with the GPU implementation was that only one block was usable as synchronization was needed between all threads, it is not in this case. Allowing the full power of the GPU to be harnessed. To further press this last point

if a second look is taken at Figures 6.17, 6.18, 6.19 and 6.20 the GPU curves are much smoother than the others. Meaning that almost all the runtime is taken up by the call to the GPU. This means that enough blocks are run with the same graph simultaneously then there could be a potential speed up compared to running them sequentially.

6.3 Discussion

In this chapter the results of the experiments are detailed. To begin an optimal ED physician schedule was produced for the weekdays and for the weekends. Following the top 100 schedules for each case where examined in order to determine how important certain shift starting points were. The results for the mincut problem were then discussed.

The physician schedules produced performed very well in regards to the metrics PIA and LOS during simulation. The key benefit that this study has over others is the shear number of schedules tested, approximately 454,000. While most other studies consider an amount of schedules on the order of 1,000 at most. This allows for the coverage a much larger problem space and the further assurance that the schedule produced is near optimal. Furthermore the key shift starting times are identified that would allow the ED to make some modifications to the schedule based on when physicians are actually able to work, due to lifestyle concerns or other obligations.

In the mincut results it was shown that the algorithm, particularly the sequential version could be reasonably integrated into the ED as part of a real time tool to aid physicians. This is due to the fact that the run time is on the order of seconds, while the time the results are need would be most likely measured in minutes. The parallelized versions however are ineffective as they do not scale very well. However some scenarios in which the parallelization could be more useful in are offered for consideration.

Chapter 7

Conclusion

7.1 Summary

In the ED physician scheduling, I was able to successfully model both the patients and the ED processes. Using the model schedules were found for both weekdays and weekends that result in a high number of patients meeting targets for PIA and LOS. If the proposed schedule was implemented and resulted in improved PIA and LOS this could result in additional funding for the ED. There were also some relationships determined between shift start times that would allow for the selection of sub-optimal schedules that better fit the other responsibility physicians have. While this simulation was developed for the TBRHSC, this technique could be applied to other EDs. This method searched a far greater problem space than is typically considered in similar studies, as can be seen in the related work section. Therefore assuring an approximation that is closer to the optimal as well as providing the additional information for choosing related schedules based on the hospitals managerial constraints. The later is not typically included in studies and provides a benefit for the practical choosing of schedules.

With respect to the minimum cut problem, the proposed algorithm performed very well compared to the gurobi solver with the ILP model. This means that it

could provide the basis for a real time tool to assist physicians in many situations potentially further improving PIA and LOS metrics for the ED. When examining parallelization of the algorithm, only the initial partitioning portion showed promise for using CPU threading to quickly solve a single case. The paralleliztion of the swapping component had too much overhead with both the CPU and GPU being outperformed by the sequential version. As mentioned it is a very busy process and GPU calls are too expensive. Alternatively, a more beneficial use of paralellization would be running multiple instances at once as the result ultimately depends on the seeding of the initial partition portion.

7.2 Future Work

In regards to future work there are several avenues that can be taken.

First, the modeling could be improved by assessing the time TBRHSC ED physicians spend with patients during different points of the patients stay. Additionally, a study could be done to determine how physicians triage which patients to see next. With proper data for these two things the simulation could be further improved.

Another avenue that could be taken would to be adjust the model to determine additional scenarios. Rather than maximizing the PIA and LOS targets, scenarios could be done to test the robustness of schedules with different patient demographics, arrival time, and volume. Alternatively, a cost benefit analysis could be done to add additional physicians to the schedule or implement physicians at triage as discussed in the related work section. With additional modifications and data it could even be used to determine the benefit of adding equipment, such as an ED specific CT scanner.

Turning towards the minimum cut problem physicians could be consulted to determine the specific scenarios in which it could be useful and determine a method for the construction of graphs to frame their problems. Additionally, comparisons could be done to determine the benefit of looking at multiple swaps before determining if the swap should be done. This could help determine if potentially obtaining a better minimum cut value is worth the additional runtime and whether parallelization becomes feasible for individual runs at this level.

This algorithm for the minimum cut problem could be implemented as the basis of a real time tool to aid staff. It could be added in several stages of a patients stay. One example would be in triage. This would allow the algorithm to be run

as the nurse enters the patients information, aiding in more accurate triaging, and providing the physician with laboratory tests and imaging procedures the patient may require. It could also provide the likely hood of a patient's stay resulting in admission. These last two scenarios would be able to reduce the patients LOS by ordering test and procedures that are extremely likely from triage, shortening time between initial assessment and reassessment. As well as prepping area in the hospital for a patient very likely to be admitted ahead of time, to reduce the time spent bed blocking. Due to the time efficiency of the algorithm these likely hoods could be calculated repeatedly to provide up to date information to physicians.

In order to use the algorithm as a basis for a real time tool further studies will need to be done heavily involving the medical community to identify proper weighted relationships between factors. Since these relationships are not ED specific it would allow the system to be easily added to any ED. All that would be needed would be an intermediate layer to translate the data to how it is represented in a graph. For example things like chief complaint may not have the same wording between hospitals.

Bibliography

[1] Y. Chai, Z. Wheeler, P. Herbison, C. Gale, and P. Glue. Factors associated with hospitalization of adult psychiatric patients: Cluster analysis. *Australasian psychiatry : bulletin of Royal Australian and New Zealand College of Psychiatrists*, 21, 02 2013.

[2] T. Chan, J. Killeen, D. Kelly, and D. A Guss. Impact of rapid entry and accelerated care at triage on reducing emergency department patient wait times, lengths of stay, and rate of left without being seen. *Annals of emergency medicine*, 46:491–7, 03 2006.

[3] D. S. Cheung, J. J. Kelly, C. Beach, R. P. Berkeley, R. A. Bitterman, R. I. Broida, W. C. Dalsey, H. L. Farley, D. C. Fuller, D. J. Garvey, K. M. Klauer, L. B. McCullough, E. S. Patterson, J. C. Pham, M. P. Phelan, J. M. Pines, S. M. Schenkel, A. Tomolo, T. W. Turbiak, J. A. Vozenilek, R. L. Wears, and M. L. White. Improving handoffs in the emergency department. *Annals of Emergency Medicine*, 55(2):171 – 180, 2010.

[4] J. Chrusciel, X. Fontaine, A. Devillard, A. Cordonnier, L. Kanagaratnam, D. Laplanche, and S. Sanchez. Impact of the implementation of a fast-track on emergency department length of stay and quality of care indicators in the champagne-ardenne region: a before–after study. *BMJ Open*, 9(6), 2019.

[5] L. G. Connelly and A. E. Bair. Discrete event simulation of emergency department activity: A platform for system-level operations research. *Academic Emergency Medicine*, 11(11):1177–1185, 2004.

[6] J. Considine, M. Kropman, E. Kelly, and C. Winter. Effect of emergency department fast track on emergency department length of stay: a case–control study. *Emergency Medicine Journal*, 25(12):815–819, 2008.

[7] M. W. Cooke, S. Wilson, and S. Pearson. The effect of a separate stream for minor injuries on accident and emergency department waiting times. *Emergency Medicine Journal*, 19(1):28–30, 2002.

[8] M. Dinh, A. Walker, A. Parameswaran, and N. Enright. Evaluating the quality of care delivered by an emergency department fast track unit with both nurse practitioners and doctors. *Australasian Emergency Nursing Journal*, 15(4):188 – 194, 2012.

[9] J. F. Dreyer, S. L. McLeod, C. K. Anderson, M. W. Carter, and G. S. Zaric. Physician workload and the canadian emergency department triage and acuity scale: the predictors of workload in the emergency room (power) study. *CJEM*, 11(4):321–329, 2009.

[10] M. E Levsky, S. E Young, L. N Masullo, M. A Miller, and T. J S Herold. The effects of an accelerated triage and treatment protocol on left without being seen rates and wait times of urgent patients at a military emergency department. *Military medicine*, 173:999–1003, 10 2008.

[11] E. El-Darzi, R. Abbi, C. Vasilakis, F. Gorunescu, M. Gorunescu, and P. Millard. *Length of Stay-Based Clustering Methods for Patient Grouping*, volume 189, pages 39–56. 03 2009.

[12] O. El-Rifai, T. Garaix, V. Augusto, and X. Xie. A stochastic optimization model for shift scheduling in emergency departments. *Health Care Manage. Sci.*, pages 1–14, 01 2014.

[13] G. W. Evans, E. Unger, and T. B. Gor. A simulation model for evaluating personnel schedules in a hospital emergency department. In *Proceedings Winter Simulation Conference*, pages 1205–1209, Dec 1996.

[14] M. L. García, M. A. Centeno, C. Rivera, and N. DeCario. Reducing time in an emergency room via a fast-track. In *Winter Simulation Conference*, 1995.

[15] K. Ghanes, O. Jouini, Z. Jemai, M. Wargon, R. Hellmann, V. Thomas, and G. Koole. A comprehensive simulation modeling of an emergency department: A case study for simulation optimization of staffing levels. In *Proceedings of the Winter Simulation Conference 2014*, pages 1421–1432, Dec 2014.

[16] S. Grant, D. Spain, and D. Green. Rapid assessment team reduces waiting time. *Emergency Medicine*, 11(2):72–77, 1999.

[17] L. Green, J. Giulio, G. RA, and J. Soares. Using queueing theory to increase the effectiveness of emergency department provider staffing. *Acad Emerg Med*, 13, 01 2006.

[18] L. V. Green, P. J. Kolesar, and W. Whitt. Coping with time-varying demand when setting staffing requirements for a service system. *Production and Operations Management*, 16(1), 2007.

[19] L. C. Hampers, S. Cha, D. J. Gutglass, H. J. Binns, and S. E. Krug. Fast track and the pediatric emergency department: Resource utilization and patient outcomes. *Academic Emergency Medicine*, 6(11):1153–1159, 1999.

[20] J. H. Han, D. France, S. Levin, I. D Jones, A. B Storrow, and D. Aronsky. The effect of physician triage on emergency department length of stay. *The Journal of emergency medicine*, 39:227–33, 02 2009.

[21] S. Hao, B. Jin, A. Shin, Y. Zhao, and C. Zhu. Risk prediction of emergency department revisit 30 days post discharge: A prospective study. *PLoS ONE*, 9, 11 2014.

[22] L. B. Holm and F. A. Dahl. Simulating the effect of physician triage in the emergency department of akershus university hospital. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 1896–1905, Dec 2009.

[23] S. Ieraci, E. Digiusto, P. Sonntag, L. Dann, and D. Fox. Streaming by case complexity: Evaluation of a model for emergency department fast track. *Emergency Medicine Australasia*, 20(3):241–249, 2008.

[24] J. Imperato, D. Scott Morris, D. Binder, C. Fischer, J. Patrick, L. Sanchez, and G. Setnik. Physician in triage improves emergency department patient throughput. *Internal and emergency medicine*, 7:457–62, 08 2012.

[25] G. D. Innes, R. Stenstrom, E. Grafstein, and J. M. Christenson. Prospective time study derivation of emergency physician workload predictors. *Canadian Journal of Emergency Medicine*, 7(5):299–308, 2005.

[26] H. K Simon, D. Mclario, R. Daily, C. Lanese, J. Castillo, and J. Wright. "fast tracking" patients in an urban pediatric emergency department. *The American journal of emergency medicine*, 14:242–4, 06 1996.

[27] A.-M. Kelly, M. Bryant, L. Cox, and D. Jolley. Improving emergency department efficiency by patient streaming to outcomes-based teams. *Australian Health Review*, 31(1):16–21, 2 2007.

[28] D. King, D. Ben-Tovim, and J. Bassham. Redesigning emergency department patient flows: Application of lean thinking to health care. *Emergency medicine Australasia : EMA*, 18:391–7, 09 2006.

[29] A. Komashie and Ali Mousavi. Modeling emergency departments using discrete event simulation techniques. In *Proceedings of the Winter Simulation Conference, 2005.*, pages 5 pp.–, Dec 2005.

[30] F. L. Counselman, C. A. Graffeo, and J. T. Hill. Patient satisfaction with physician assistants (pas) in an ed fast track. *Elsevier*, 18(6):661–665, 10 2000.

[31] K. L. Murrell, S. R. Offerman, and M. B. Kauffman. Applying lean: implementation of a rapid triage and treatment system. *The western journal of emergency medicine*, 12(2):184–191, 5 2011.

[32] W. Liu, Z. Wang, X. Liu, W. Yue, and D. Bell. A clustering approach to triage categorization in a e departments. In *2017 23rd International Conference on Automation and Computing (ICAC)*, pages 1–6, 2017.

[33] B. C. Maughan, L. Lei, and R. K. Cydulka. Ed handoffs: observed practices and communication errors. *The American journal of emergency medicine*, 29 5:502–11, 2011.

[34] M. MD, M. MD, M. MD, M. Gorelick, E. Alpern, and E. Alessandrini. A system for grouping presenting complaints: The pediatric emergency reason for visit clusters. *Academic Emergency Medicine*, 12:723 – 731, 08 2005.

[35] S. N. Partovi, B. K. Nelson, E. D. Bryan, and M. J. Walsh. Faculty triage shortens emergency department length of stay. *Academic emergency medicine : official journal of the Society for Academic Emergency Medicine*, 8:990–5, 11 2001.

[36] D. O'Brien, A. Williams, K. Blondell, and G. A. Jelinek. Impact of streaming fast track emergency department patients. *Australian Health Review*, 30(4):525 – 532, 2006.

[37] V. Prasad, J. C. Lynch, M. R. Filbin, A. T. Reisner, and T. Heldt. Clustering blood pressure trajectories in septic shock in the emergency department. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 494–497, 2019.

[38] V. Quattrini and B. Swan. Evaluating care in ed fast tracks. *Journal of emergency nursing: JEN : official publication of the Emergency Department Nurses Association*, 37:40–6, 01 2011.

[39] B. R Holroyd, M. Bullard, K. Latoszek, D. Gordon, S. Allen, S. Tam, S. Blitz, P. Yoon, and B. H Rowe. Impact of a triage liaison physician on emergency department overcrowding and throughput: A randomized controlled trial. *Academic emergency medicine : official journal of the Society for Academic Emergency Medicine*, 14:702–8, 09 2007.

[40] J. R Richardson, G. Braitberg, and M. J Yeoh. Multidisciplinary assessment at triage: A new way forward. *Emergency medicine Australasia : EMA*, 16:41–6, 03 2004.

[41] C. Rapeli and N. Botega. Clinical profiles of serious suicide attempters consecutively admitted to a university-based hospital: A cluster analysis study. *Revista brasileira de psiquiatria (São Paulo, Brazil : 1999)*, 27:285–9, 01 2006.

[42] M. Raunak, L. Osterweil, A. Wise, L. Clarke, and P. Henneman. Simulating patient flow through an emergency department using process-driven discrete event simulation. In *2009 ICSE Workshop on Software Engineering in Health Care*, pages 73–83, May 2009.

[43] S. W. Rodi, M. V. i Grau, and C. M. Orsini. Evaluation of a fast track unit: alignment of resources and demand results in improved satisfaction and decreased length of stay for emergency department patients. *Quality management in health care*, 15 3:163–70, 2006.

[44] T. Rogers, N. Ross, and D. Spooner. Evaluation of a 'see and treat' pilot study introduced to an emergency department. *Accident and emergency nursing*, 12:24–7, 02 2004.

[45] M. D. Rossetti, G. F. Trzcinski, and S. A. Syverud. Emergency department simulation and determination of optimal attending physician staffing schedules. In *WSC'99. 1999 Winter Simulation Conference Proceedings. 'Simulation - A Bridge to the Future' (Cat. No.99CH37038)*, volume 2, pages 1532–1540 vol.2, Dec 1999.

[46] M. Rouzbahman, A. Jovicic, and M. Chignell. Can cluster-boosted regression improve prediction: Death and length of stay in the icu? *IEEE Journal of Biomedical and Health Informatics*, 21:1–1, 02 2016.

[47] S. Russ, I. Jones, D. Aronsky, R. S Dittus, and C. M Slovis. Placing physician orders at triage: The effect on length of stay. *Annals of emergency medicine*, 56:27–33, 03 2010.

[48] M. Sanchez, A. J. Smally, R. J. Grant, and L. M. Jacobs. Effects of a fast-track area on emergency department performance. *The Journal of Emergency Medicine*, 31(1):117 – 120, 2006.

[49] D. Savage, D. G Woolford, B. Weaver, and D. Wood. Developing emergency department physician shift schedules optimized to meet patient demand. *CJEM*, 17:3–12, 03 2015.

[50] S. Y. Shin, H. Balasubramanian, Y. Brun, P. L. Henneman, and L. J. Osterweil. Resource scheduling through resource-aware simulation of emergency departments. In *2013 5th International Workshop on Software Engineering in Health Care (SEHC)*, pages 64–70, May 2013.

[51] D. Sinreich and O. Jabali. Staggered work shifts: A way to downsize and restructure an emergency department workforce yet maintain current operational performance. *Health care management science*, 10:293–308, 10 2007.

[52] D. Sinreich, O. Jabali, and N. P. Dellaert. Reducing emergency department waiting times by adjusting work shifts considering patient visits to multiple care providers. *IIE Transactions*, 44(3):163–180, 2012.

[53] D. SINREICH and Y. MARMOR. Emergency department operations: The basis for developing a simulation tool. *IIE Transactions*, 37(3):233–245, 2005.

[54] G. Soler, G. Bouleux, E. Marcon, A. Cantais, S. Pillet, and O. Mory. Emergency department admissions overflow modeling by a clustering of time evolving clinical diagnoses. In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pages 365–370, 2018.

[55] C. R. Standridge. A tutorial on simulation in health care: applications and issues. In *WSC'99. 1999 Winter Simulation Conference Proceedings. 'Simulation - A Bridge to the Future' (Cat. No.99CH37038)*, volume 1, pages 49–55 vol.1, Dec 1999.

[56] F. Subash, F. Dunn, B. McNicholl, and J. Marlow. Team triage improves emergency department efficiency. *Emergency medicine journal : EMJ*, 21:542–4, 09 2004.

[57] A. Tabaie, F. Chokshi, A. Holder, and S. Nemati. Doubly-robust estimation of effect of imaging resource utilization on discharge decisions in emergency departments. volume 2018, pages 3256–3259, 07 2018.

[58] J. Terris, P. Leman, N. O'Connor, and R. Wood. Making an impact on emergency department flow: Improving patient processing assisted by consultant at triage. *Emergency medicine journal : EMJ*, 21:537–41, 09 2004.

[59] A. K. Venkatesh, D. P. Curley, Y. Chang, and S. Liu. Communication of vital signs at emergency department handoff: Opportunities for improvement. *Annals of emergency medicine*, 66 2:125–30, 2015.

[60] C. Yang, C. Delcher, E. Shenkman, and S. Ranka. Clustering inter-arrival time of health care encounters for high utilizers. In *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–6, 2018.

[61] K. Ye, D. Taylor, J. C Knott, A. Dent, and C. Macbean. Handover in the emergency department: Deficiencies and adverse effects. *Emergency medicine Australasia : EMA*, 19:433–41, 10 2007.

[62] H. Yoshida, L. E. Rutman, J. Chen, M. L. Shaffer, R. T. Migita, B. K. Enriquez, G. A. Woodward, and S. S. Mazor. Waterfalls and handoffs: A novel physician staffing model to decrease handoffs in a pediatric emergency department. *Annals of Emergency Medicine*, 73, 10 2018.