Simulating Patient Flows in a Cancer Clinic

Christian Lejdström
ine15@cle.student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisor: Emma Fitzgerald

Examiner: Christian Nyberg

June 14, 2020

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

Planning resources in cancer care is difficult due to system complexity, varying resource availability and uncertain demand. It has become so complex that the analysis of treatment processes in cancer clinics cannot be done analytically [12]. Therefore, in this thesis a model of a cancer clinic is developed using discrete event simulation. The goal of the thesis is develop the model so that it can be used as research tool to investigate bottlenecks, allocation of resources and system properties.The model will also serve as a base plate for further investigations. Most other studies on this subject have not in detail presented how the simulation handles the discrete effect of weekends and servers only being active during certain part of the day, which is addressed in thesis. In this thesis a sensitivity analysis is carried out for a number of different parameters. First we investigate the effect of increasing the load and analyse how schedule utilization and patient system time (PST) are affected. Secondly, the system sensitivity to disturbances such as sick doctors and patient rebooks are investigated. Furthermore, bottlenecks and schedule alignment is also investigated.

Glossary and Abbreviations

DES - Discrete event simulation

PST - Patient system time
Total time patient spent in the workflow

PAR - Average patient arrivals per week
In this thesis all arrivals follow a poisson distribution

MST - Mean patient system time

MGST10 - Mean of 10% greatest system times

OIS - Oncology information system

Popular Science Summary

During the past 50 years health care and cancer treatment have experienced a tremendous development. One major side effect of this is that the treatment process and healthcare clinics in general have become much more complicated. Therefore, it has become difficult to analyse inefficiencies such as bottlenecks in the treatment process. It has also become harder to study cause and effect relationships with conventional tools such as spreadsheet and organization charts. Furthermore, the complexity level also makes it difficult to study the system with analytical tools such as optimization or queuing theory. To be able to study the system a computer simulation model is developed. The model basically tries to mimic and take in all the aspects, rules and processes of a real world clinic. Of course this is not possible so therefore there are some limitations set on the model. The model is built upon on the theory of discrete event simulation. Discrete event simulation is a convenient simulation tool to use when a discrete process is to be modeled, in this model we are dealing with patient and time schedules which are discrete and therefore discrete event simulation is suitable to use for this model.

All patients that come into a clinic go through a workflow, depending on the cancer type, stage and many other factors. A workflow is a treatment process with some predefined stations such as a doctor consultation, CT-scan and radiation treatment [6]. In this thesis we only study one workflow in a radiation therapy clinic. The main goal of this thesis is to explore how the system behaves when certain parameters are changed. One of the properties investigated is how the time a patient spends in the system changes when either more or less patients enter the clinic. Another factor that is investigated is how the system reacts to disturbances. A disturbance can be that a doctor becomes sick or that a patient reschedules his or her appointment. Finally, all these interactions and results are analyzed to see if the system has some general properties that might apply to other workflows and clinic setups as well.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Company and Background

RaySearch was founded in the year 2000 by Johan Löf. It started out as a PhD project and spin off from Karolinska Institutet in Stockholm. RaySearch develops software for cancer treatment with a specialization in radiotherapy. RaySearchs main goal is to advance cancer treatment through pioneering software. Its products are used by thousands of clinics around the world in their fight against cancer. Its main products are RayStation and RayCare. RayStation is a software tool used for treatment planning in radiation oncology. An example of usage is radiation therapy treatment plan optimization and dose calculation.

RayCare is an OIS used by numerous cancer clinics around the world. RayCare is designed with the intent to support comprehensive cancer care and be the leader in the OIS market. It has many state of the art features such as rule-based scheduling, digital oncology workflow and many more. The main reason behind this thesis project is to gain useful insight into how a clinic can be modelled and what this implies for scheduling. A secondary goal is also to in the future be able to use insights and conclusions from this thesis in building even better scheduling functions and improve RayCares capabilities.

1.2 Motivation and Goals

Cancer treatment today is very costly and resource intensive, especially if a patient undergoes radiotherapy. A clinic today is very complex and there are a lot of different resources and workflows in a cancer clinic to take into consideration when planning care. Furthermore, increased system complexity, varying resource availability and uncertain demand make this process even more complicated.This makes the system difficult to study from an analytical perspective.

To get a better understanding of the system a discrete event simulation model is developed. The model will be used as a research tool to get a deeper understanding of how the system works. Some main factors to investigate are bottlenecks in clinics, the effects of alleviating bottlenecks and how best to optimize the allocation of resources in the clinic so that patient system time (PST) is minimized. Furthermore, disturbances are also investigated such as patient cancelling appointments and sick doctors.

The main goal of this thesis is to get a better understanding of how the total time a cancer patient spends in the clinic is affected by varying resources, schedules and arrival times and to figure out which of the factors that are of higher importance. Finally we also investigate how schedule utilization and variance is affected by the different experiments.This is done to see if there are any general properties that this system might exhibit.

1.3 Thesis Structure

There have been some related studies in this field which are discussed in Chapter 2. In Chapter 3 the basics of a discrete event simulation and radiotherapy are discussed.This is done in order for the reader to grasp all concepts used in building the model in Chapter 4. Chapter 4 describes the model used throughout the simulation. Although no code is presented in the thesis, it is described to such an extent that it should be possible for the reader to reproduce the simulation based on our description. Furthermore the model is described in such a way that it should be easy to understand the experiments made in Chapter 5. The experiments made in Chapter 5 investigate how the system behaves under different settings where we mainly investigate schedule utilization and PST. In Chapter 6 the results are summarized and discussed. All the results are presented in the form of figures. In Chapter 7 the most important conclusions and insights from the experiments and simulations are presented.

1.4 Contribution

In most other papers there is no real explanation of how the schedule is constructed, they do not mention if there are arrivals or treatments on weekends. Therefore this aspect is always taken into consideration in this thesis work.

A range of studies have been made on simulating cancer clinics, but none of the studies have presented a thorough model that can easily be implemented by the reader. This thesis on the other hand provides a comprehensive description of the model and methods used to build the model, which makes it easy for the reader to replicate all experiments. Furthermore, a lot of different aspects of scheduling have been presented in similar papers but there is no thorough sensitivity analysis with respect to load and disturbance. In this thesis we investigate how utilization and PST are affected by various factors such as patient arrival rate(PAR). Through comprehensive measurements clear relations are established about how PST increases non-linearly when schedule utilization is high. Additionally, it provides a detailed description of how schedule variance decreases with increased system load. Additionally, this thesis showed that even though capacity is the same throughout all activities in a schedule it is hard to predict were the bottlenecks are. Finally, the disturbance experiment indicated that only cancellation effects have an impact on PST and schedule variance.

1.5 Limitations

Since the main focus of this thesis has been on model development no validation of model output with real world clinic data has been made. This is due to the fact that the model constructed is quite simplified compared to a real word clinic which makes a comparison difficult. In a normal clinic there are also a lot of random events, some are captured in this model but not all. To capture the random events to some extent a selected few have been implemented as disturbances.

Chapter 2

Related Work

2.1 Related Studies

A vast majority of studies have been made in the area of improving the effectiveness and resource management in hospitals and clinics using a range of methods including discrete event simulation. Therefore a number of studies was chosen that most closely resembles this thesis. These studies also served as baseplate for what to investigate further.

2.1.1 Scheduling Strategies

A study made by [12] uses discrete event simulation to study the impact of using a pull versus a push strategy when scheduling. The goal of the study is to try to improve the effectiveness of the scheduling strategy. The simulation was carried out with following events: CT, MRI, PET-CT, Consultation, Contouring, Image post processing, Treatment Planning and fractions at Linac. The treatment workflow in this study is shown in Figure 2.1. A pull strategy implies that the start of treatment is set right after the consultation, which in this case is the start of the first fraction. This demands that the pre-treatment workflow is scheduled after the first fraction. A push strategy implies that the pre-treament workflow is scheduled on a more continuous basis and in time chronological order. In our study a push strategy is also used i.e scheduling is made on a more continuous basis. Our workflow used in this study is based upon [12] paper. Although a similar workflow was used they did not mention more precisely how the interaction with scheduling was constructed when utilizing a push strategy. Additionally, the study does not mention anything more in detail regarding how specific time slots are distributed over the weeks, for doctors and machines.

Most of the studies use a Poisson process to model arrival processes including this one. Our decision to use a Poisson arrival process was based upon this fact.

[4] investigates how to design an optimal doctor's schedule with the goal to minimize access time for patients, in an outpatient radiotherapy clinic. Access time is defined as the time it takes from referral to start of treatment (first fraction), measured in calendar days. In this thesis where we aim to study how the system behaves under different settings and loads and therefore we measure the whole

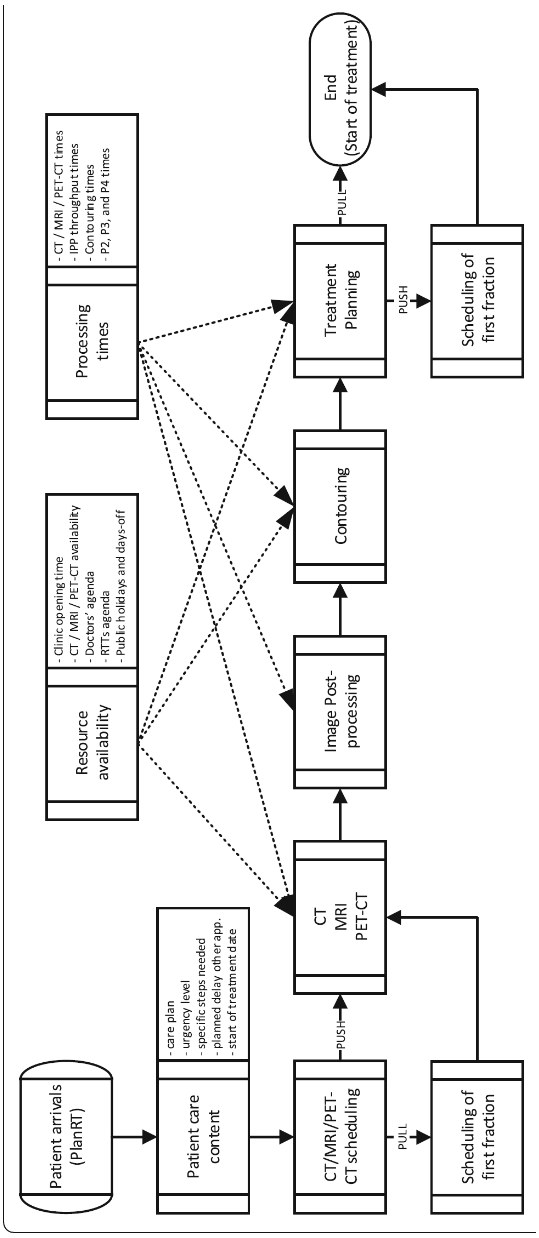

Figure 2.1: Treatment workflow - [12]

time instead as PST. One of these experiment is to rearrange server schedules similar to what they do in these experiments.

[4] try to optimize schedule based on minimal treatment time, which is not done in our experiments. In our simulation the doctors schedule is weekly cyclic as in their study. The model used is given in Figure 2.2. Our model uses DES(discrete event simulation) to capture and model all aspects of the system whereas here they have a purpose of using integer linear programming instead.

2.1.2 Variance and fluctuation reduction

One clear goal in this thesis is to investigate how schedule utilization and variance are affected, mainly because low utilization and high variance is undesirable from an economic perspective. [8] investigates which steps in the process in an outpatient radiotherapy clinic that are responsible for preventing to reach desired throughput. This is quite similar to this thesis although our approach is to study how the system behaves with varying PAR, not to reach a specific output. Furthermore the authors aim to analyze how variability reduction in the capacity affects overall effectiveness of the system. In our experiments there is no resource variability except for the disturbances, since our schedules are weekly cyclic or deterministic. There is also no mentioning in more detail exactly how the schedule and resources varies for a year, which is done in our experiments. The model used here is shown in Figure 2.3.

2.1.3 Sensitivity analysis and what if strategies

[13] constructs a discrete event simulation model for an outpatient radiotherapy clinic, given in 2.4, [2] also conduct a similar study.The main goal of theses studies is an exploratory research where different types of what if studies are carried out to find bottlenecks and get a deeper understanding of how the system works. Our thesis is also constructed as an exploratory research. The main difference is that the authors only choose a couple of factors such as arrival increase by a certain percentage or increased productivity by a factor x. Therefore these approaches do not capture the dynamics of the system fully, as it could have done if more data point were added as in our experiments.

2.1.4 Studies without DES

Studies of cancer clinics and healthcare clinics in general are made within operation research. In operations research there are many different studies investigating cancer clinic and clinics in general that do not use DES. A sample of the studies that were most similar to our study is presented in this section.

[11] uses mathematical programming and queuing theory to try to optimize capacity resource allocation for Linacs in a radiotherapy clinic. In our study we investigate capacity allocation for doctors but the number of linacs is kept constant. [10] aims to study PST and in particular minimizing it, which is done with the use of a genetic algorithm. The genetic algorithm is used to generate schedules with good performance. Although the paper minimizes PST they also investigate

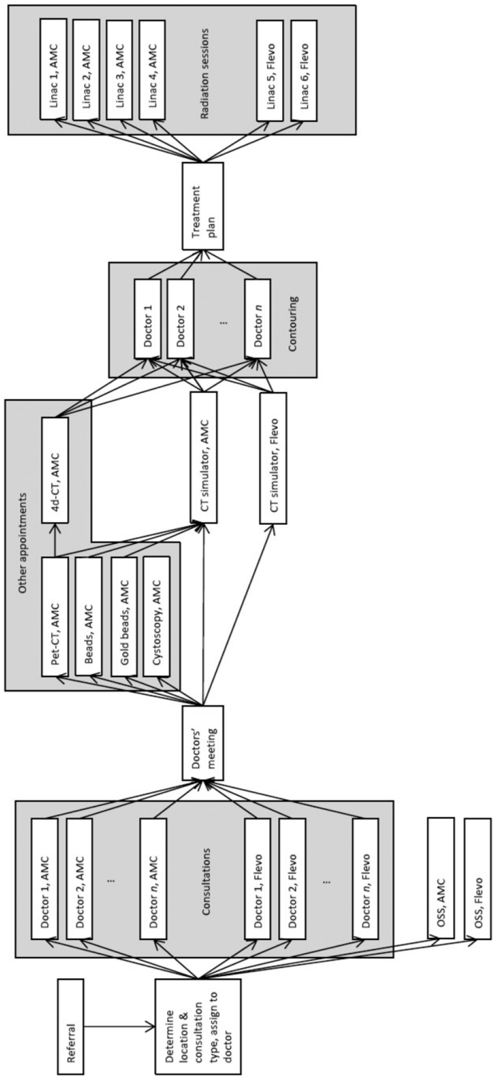

Figure 2.2: Setup for discrete event simulation - [4]

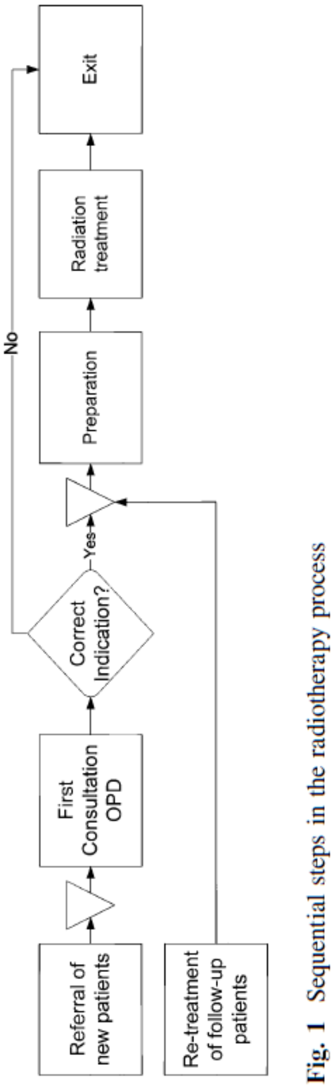

Figure 2.3: Setup for discrete event simulation - [8]

it as function of different schedules as in this thesis. The authors use this thesis to show benefits to clients. [5] uses mixed integer linear programming to improve treatment scheduling. Unlike our thesis patient have different priorities but the overall goal is to minimize PST and maximize linac utilization. With the MLP algorithm implemented they managed to improve utilization. Data used is only fictional like in our experiments. [9] also uses mathematical programming to study a radiotherapy system. The author introduces multiple treatment patterns that is similiar to the workflow of this thesis. It is done in order to model the clinic more realistically. They only use fictional generated patient data.

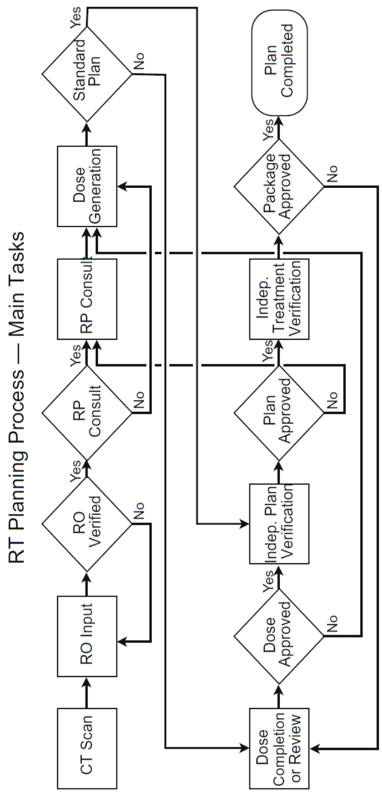

Figure 2.4: Setup for discrete event simulation - [13]

Chapter 3

Technical Background

This chapter introduces key concepts to the reader about discrete event simulation. The purpose of explaining these is so that the reader more easily can understand how the model is built in Chapter 4. Furthermore, there is also a brief overview about radiotherapy so that the reader can get a better understanding of the workflow, treatment process and model.

3.1 Radiotherapy

A Swedish book by Jan Degerfalt presents a thorough overview over the radiotherapy treatment process [6]. Radiotherapy is a type of cancer treatment used to treat tumors. It is often used as a stand alone cancer treatment or in combination with other methods such as chemotherapy. There are mainly two types of purposes for treatment: palliative and curative. Palliative treatment aims to decrease the amount of symptoms the cancer gives. The main purpose is not to prolong the patients life but instead to increase quality of life. Curative treatment on the other hand aims to cure the patient fully. This does often imply that certain side effects will be present. The main cause of this is the higher doses a patient receives in a treatment. Treatment is made with certain types of machines that deliver ionizing or non ionizing radiation.

3.1.1 Treatment delivery

The most common form of radiation therapy is external beam radiotherapy. This means that treatment is carried out with an external radiation source with ionizing radiation [6].The types of radiation use can be divided into two groups photons and ion particle radiation. Treatment is delivered in fractions. A fraction is defined as one radiation treatment session. Typically a patient in a curative treatment receives between 1 and 38 fractions. Normally a patient is treated with one or two fractions per day. The radiation treatment is most commonly delivered on a linear accelerator (Linac) in the case of treatment with photon and otherwise on a particle accelerator.

3.1.2 Treatment process

Radiotherapy is often quite complex. There a are number of steps involved before the actual treatment is carried out. Since there are many cancer types the treatment process can vary greatly from patient to patient . A typical treatment process is presented in figure 2.1. Instead of the full range of possible treatment steps, we present a select few that are the most common ones[6]:

Consultation: The first step in the treatment process where a patient is referred from a general clinic and has the first meeting with a radiation oncologist.

CT scan: Computed tomography scan (CT) is a medical imaging procedure, that uses x-rays to create cross section images of selected parts of the body. The main purpose of the scan is to identify and analyse different densities of the area that is to be treated

Target contouring: From the images created in the CT-scan, journals and other data a target volume and radiation dose is defined. Target volume is the shape/volume that is to be treated.

Image processing: To extract and compile images from different types of scan and x-rays they are often processed with different types of algorithms and programs.

MRI: Magnetic resonance imaging (MRI) is a type of imaging technique that can be used to generate images. It can be used as a complement to CT-scans or as a replacement.

PET CT - scan: PET-CT is a combined medical imaging procedure consisting of a CT and Position emission tomography (PET) that is done simultaneously on a single machine.

Plan review: Doctor/physician reviews plan to ensure there are no errors in the plan. If there is they are corrected

3.2 Queuing Theory

Queuing theory deals with how to mathematically describe queues [1]. It involves all aspects of queuing: arrival processes, service processes, number of servers and other parameters. Queuing theory can also be used to study and gain useful insights into discrete event systems. It can also be viewed as a by hand alternative to discrete event simulation. Since queuing theory has a lot of analytical solutions to different queuing systems it can provide some powerful tools for simulation models:

- *Validation* is used as tool to validate part of a complicated simulation model to ensure results are correct or reasonable
- *Insight* by studying the analytic models it is more easy to get useful insight into complex systems.

A queuing system is described by using the Kendall notation: A/S/c, where A denotes the arrival time between queues, S the service time distribution and c the number of servers that are available. The arrival process in a queuing system describes how a patient arrives according to a distribution and with a certain rate [1]. This can either be a defined analytic distribution such as the Poisson distribution or it can be experimental. The service time distribution S is specified in the same way as the arrival distribution.

3.2.1 Steady state

Not all queuing systems have a steady state, but some do. Steady state theory focuses on the queuing system in the long run [7]. This is a main concern because there is a potential for massive or even infinite queues in the long run, so it is important to investigate steady state properties. Many systems also have a warm up period called the transient phase before they stabilize so the simulation cannot be made too short because then you might miss capturing the steady state.

3.3 Discrete Event Simulation

Discrete event simulation is a modelling tool that is widely used throughout many disciplines to model complex systems. Simulation in general is often used when a system is to complex to be studied analytically. A discrete event simulation is built up of multiple components [3, 7]:

Model: Abstract representation of the system. Contains structural, mathematical or logical relationships that describes the system in terms of state, entities, events and corresponding attributes

Event: The main part of the simulation, when an event is triggered the system states are updated according to the instructions specified in the event. Between two events the system does not change at all. In general an event triggers a new event (often of the same type) that is put into the event list. When an event is triggered it also updates the simulation clock.

Clock: The simulation clock in a discrete event simulation is not continuous. Time is also not real time only simulated which means that simulations can be executed much faster. Time is updated by a triggered event with a specified amount of discrete time units. Time is global and shared by all events.

Event list: The event list is the list where all events are stored before they are triggered. Events are sorted according to their trigger time. If two events have the same trigger time the one that was sent to the list first is executed first.

Entity: A component in the system that requires explicit representation. It can be a server, customer, machine etc.

Attributes: Properties of the given entity (e.g. priority of a patient)

Activity: Duration of time a specified length(e.g. a doctor appointment with fixed time).

Delay: A duration of time of indefinite length, which is not known until it ends (customer delay in waiting line).

System State: Collection of variables that can be used to describe the system at any time. Consists of queue length, servers, persons in system etc.

A visual representation of a discrete event simulation is presented below in figure 3.1.

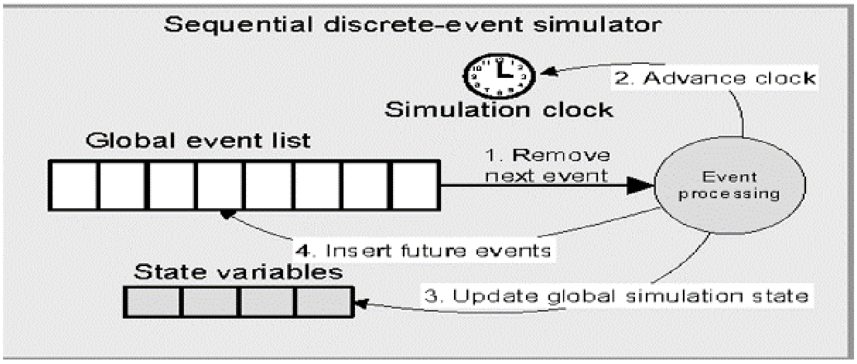

Figure 3.1: Schematic description of DES model

Chapter 4

Model

The model described here is used through all experiments. Not all parts and features of the model are used in all experiments and some may therefore be inactive. A thorough description of the model implementation is presented, to make it possible to reproduce this model and get a better understanding of the ideas and principles used. The sections and subsections of this chapter are based on the program structure.

4.1 Program Structure

The program is implemented with an object oriented design. By doing this it is effortless to run multiple simulations with different settings smoothly, because it is easy to instantiate objects with new settings. It also gives a good logical structure to the program. Below are all program modules stated, they are then explained more in depth throughout this chapter.

Patient Simulation: Main class, the program is executed from here. All events are implemented in this module and settings are specified here.

Event handler: Contains the description of an event and event list. All distributions are also specified here.

Schedule handler: Contains a schedule class that describes the schedule and all necessary variables.

Stats: Contains functions, methods and variables that make it possible to process and store all the necessary statistics

Entities: Contains descriptions of all our entities: doctor, machines and patients

Algorithms: Contains all algorithms used in this experiment.

4.1.1 Module interaction

In figure 4.1 the object oriented model is presented in more detail. This is done in order for the reader to get a better overview over the model. Although it is presented in more detail all methods relations and attributes are not included, only a sample is selected to ensure that the basic implementation of the model can be understood. From the Figure we can see that when an event is triggered a booking process is initiated which is described in more detail throughout this chapter. Furthermore, if a measurement event is triggered measurement attributes are updated as can be seen in the Figure and at the end of the simulation all statistics are calculated. From the Entities module we can see that the entity is stored within the event and that each event is stored in the event list.

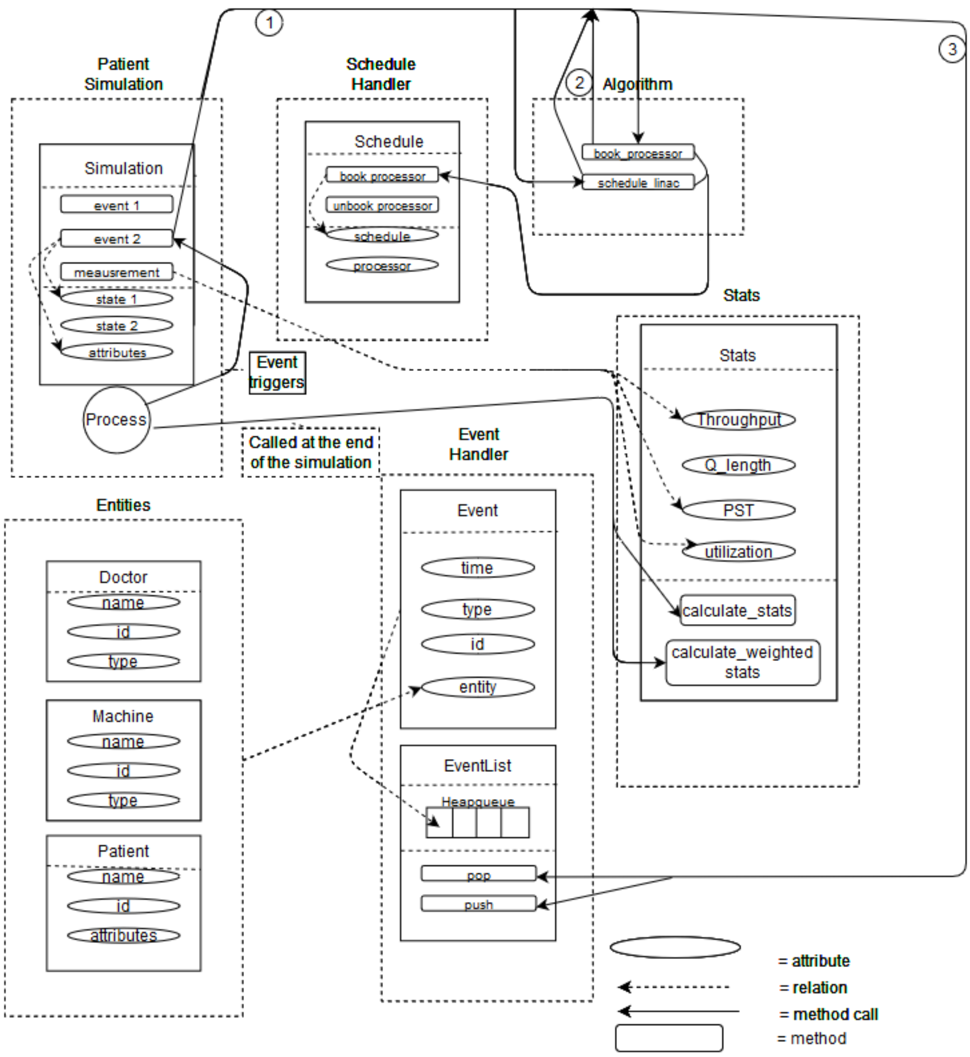

Figure 4.1: Model interaction

4.2 Patient Simulation

The main part of the program which has only one class: Simulation. Putting the simulation in a class makes for a convenient way to instantiate a simulation with different settings.

4.2.1 Simulation states

A large number of states is needed to describe the system correctly. Without the states defined properly the simulation does not run. The main state categories are schedules, queues, system time and the event list. Many important states are also stored inside our entities, mainly the patients. Measurement variables such as throughput per day are needed in order to measure important variables but they are not needed in order for the simulation to run.

4.2.2 Events

Below is a table of all events used in the different experiments in table 4.1. There are 3 different categories:

Normal: Events that are standard which are when a patient is scheduled for treatment or when a doctor does a certain task.

Measurement : Events that keep track of statistical variables and measurements at regular intervals such as: utilization, throughput and queue length.

Disturbances: Events that interrupt or disturb the process. It can be events that do rebooking of a patient or that a doctor become sick.

Triggers describes which new event is inserted into the event heap when the current event triggered. An event can either trigger zero or more events. Typically it triggers one event.

Disturbances

Disturbances are events that represent a situation where it does not go as planned. Disturbances can be categorized into two parts; *rebooking* and *cancellation*. Rebooking is when an old appointment is canceled and made available for another event and a new slot is booked on a different day. Disturbances that belong to this category are: patient rebook, doctor rebook and linac rebook. Cancellations are disturbances where the current appointment is canceled and rebooked, but the old slot is made unavailable for new bookings.

4.2.3 Booking methods

Due to the different nature of many events such as rebook and unbooking, booking methods were implemented to simplify the program structure. The main difficulty was that linac scheduling was done in batches whereas scheduling for other activities were done concurrently. Therefore a total of four booking methods were implemented:

Event	Category	Triggers	Description
Patient arrival	Normal	Patient arrival Consultation	Arrival event
Consultation	Normal	CT-scan	Doctors appointment patient present
CT-scan	Normal	Contouring PET-CT MRI	Patient scanned in machine One of 3 triggered randomly
PET-CT	Normal	Contouring	Patient scanned in machine
MRI	Normal	Contouring	Patient scanned in machine
Contouring	Normal	Plan review	Made by doctor patient not present
Plan review	Normal	Start fraction	Made by doctor patient not present
Start fraction	Normal	Receive fraction	Patient in Linac
Receive fraction	Normal	None	Patient in Linac
Sick doctor	Disturbance	Sick doctor	Doctor get sick for a day Patients are rebooked
Patient rebook	Disturbance	Patient rebook	Either a Consultation or CT-scan is rebooked
Doctor rebook	Disturbance	Doctor rebook	Either a Contouring or plan-review is rebooked
Linac rebook	Disturbance	Linac rebook	A slot on a linac machine is rebooked
Throughput	Measure	Througput	Called once a day measures throughput
Utilization	Measure	Utilization	Measure schedule utilization

Table 4.1: Event list model

Unbook: Unbook all activities from a given schedule except linac. This method is only used by disturbance events and therefore draws a random patient from the queue specified as an input parameter. Can be used to unbook any activity in a given schedule as long as it is not booked in batches. The given event is not removed from the event list, therefore it is added to a blocked event list to be sure it does not trigger. This goes for all slots that are unbooked.

Rebook: Rebooks the unbooked patient on any day that is not the same as the unbooked day.

Book Linac: Used for batch booking. Books all specified number of fractions at once according to the specified algorithm.

Unbook Linac: Unbooks the patient from linac. Constructed as a separate method because multiple operations need to be done in order to remove a slot correctly. Can be used to unbook all activities that are batch booked.

Book processor: Generic booking method that can be used to book a slot in any schedule.Has no support for batch booking.

4.3 Event Handler

This module handles all operations related to generating events, describing events and the event-list. The properties event, event list and event generator is described by their respective classes.

4.3.1 Event list

The event list used in this model is a heap queue. The reason why a heap queue was chosen is that fast sorting is possible increasing the simulation speed. In total the event heap has two functions *pop* and *push*. These two functions are used to remove and insert events from the heap queue. Information in the heap queue is stored as a tuple with the event time and corresponding event. If two events have the same time they are sorted based on event id.

A hash set with blocked events was also incorporated. An event is added to the blocked event set when a disturbance event is triggered. Because the old event shall not trigger (that is rebooked or canceled), it is added to a list of blocked events instead of being removed from the heap queue.

4.3.2 Event

Event stores all information needed to describe a particular event. The event objects are then stored in the event list until they are triggered. Each event is described by the following attributes:

Time: Stores the time at which the event should be triggered.

Id: The event id is unique for each event it is used as a key to keep track of a specific event.

Type: Keeps track of which type an event belongs to. Event type is a numerical representation that tells the event call function which method to call in the simulation class.

Entity: Not every event has an entity, for example measurement events do not. Almost all events have the patient as an entity. Mainly because all almost activities are based on state change related to the patient.

4.3.3 Event Generator

Event generator is a helper class that is used for generating random number from certain distributions. In total it supports four distributions: normal, gamma, exponential and Poisson.

4.4 Algorithm

The algorithm module is constructed in such a way that it is easy to switch out the algorithm. Currently one algorithm is implemented and it schedules greedily. A description of the algorithm is presented below in Algorithm 1. The algorithm starts the iteration from the current day, it then checks if a patient has the same server throughout the workflow. If that is the case it only checks the specific server for the first available slot and books it. Otherwise it loops through each server consecutively to find the first available slot.

Algorithm 1 Greedy Booking algorithm

```
procedure SCHEDULE SERVER(schedule, patient, current_day, same_server, server)
    for day_schedule in schedule[current_day :] do
        if same_server then
            for slot in server do
                if slot is available then schedule.book(slot, patient)
                    return
                end if
            end for
        end if
        for server in day_schedule do
            for slot in server do
                if slot  in then schedule.book(slot, patient)
                    return
                end if
            end for
        end for
    end for
end procedure
```

4.5 Schedule Handler

Contains the schedule class that describe a generic schedule. The time scale here is not continuous i.e in some intervals there are restrictions if an event can take place or not. This is because machines and doctors only operate during weekdays, whereas the simulation runs over a whole year. Therefore most normal events need to take place in specified schedules and timeslots. To account for this in the model there has to be an interaction between the schedule and the event list in order for the simulation to work as intended. This interaction is shown in figure 4.2.

In general one slot corresponds to 30 minutes therefore no activity is assumed to be shorter than 30 minutes. The only exception is the linac where one slot is 10 minutes.

The schedule module consists of a schedule class that describes a schedule. Schedules are implemented using a nested list. A typical schedule is presented in figure 4.3. In each schedule slot there are five pieces of information stored:

Time: Time of the slot stored as a date time object

Server: Server that belongs to the slot. For example a doctor.

Scheduled: Boolean variable that tells if the server is scheduled or not

Activity: Specifies which activityis allocated to the slot, if the server is not scheduled the value is set to None.

Patient: Specifies who is treated in the current activity.This does not mean that the patient has to be present in the clinic.

Status: Specifies if the doctor is sick or not.

The schedule also has a number of methods in order to perform operations on it. These are: book_server, unbook_server, get_utilization and clear_day_schedule.

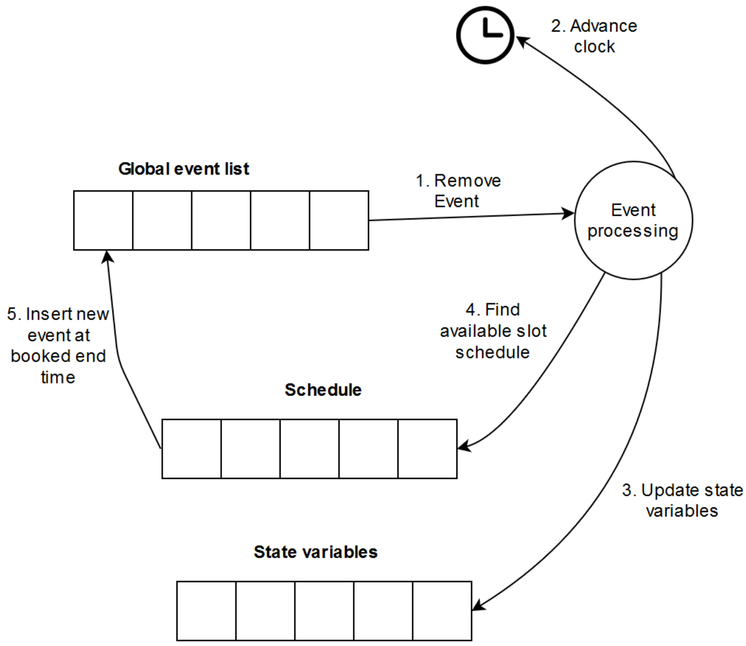

Figure 4.2: Discrete Event Simulation with Schedule

Doctor	08:00-08:30	08:30-09:00	09:00-09:30	09:30-10:00	10:00-10:30	10:30-11:00	11:00-11:30	11:30-12:00	12:00-12:30	12:30-13:00
Charles	Not scheduled None	Not scheduled None	Not scheduled None	Not scheduled None	Not scheduled None	Not scheduled None	Not scheduled None	Not scheduled None	Not scheduled None	Not scheduled None
Allan	Booked P: 615 consultation	Booked P: 615 consultation	Booked P: 616 consultation	Booked P: 616 consultation	Booked P: 617 consultation	Booked P: 617 consultation	Booked P: 618 consultation	Booked P: 618 consultation	Booked P: 620 consultation	Booked P: 620 consultation

Figure 4.3: Example of schedule

4.6 Statistics and Measurements

In queuing theory and simulation in general it is important to study the stationary or steady state of the system. Not all simulations will have a clear steady state. But in this model the measurements of all variables is constructed so that they are taken after the warm up period, if there is one. This was ensured by doing a long simulation run to see when the system on averaged reached it's steady state.

The measurements are carried out once a day for all variables that are not absolute, such as the time a patient spends in the system. The measurement is taken at 23.59 each day, by doing this it is made sure that all activities that happen during the day are included in the measurement.

In order to decrease the variance of the results a total of 20 runs is made. To calculate an average value the mean is taken over 20 different simulations for each day after the warm up period. Below is a table of everything that is measured in the simulation:

Queue length: Length of all queues to the servers at the end of the day

Utilization: The utilization of each schedule is measured. Utilization is defined as: $used\, slots/scheduled\, slots$.

Patient time in system: The patient time in system is defined as the time from when the patient first comes in for a consultation to the time when the patient is discharged after the last fraction.

Throughput: Throughput is measured for each server. It is defined as the number of patient that pass through each server on a given day.

The statistic module is constructed to handle the collection of all data. From these four measurement variables variance, confidence interval and frequency table are calculated. All data is stored in csv file format to be accessible at a later time.

4.7 Entities

The entities module contains classes describing all our entities. Currently these are machine, doctor and patient. The reason why CT and MRI are not described as separate entities is because they can be described by a general machine object. Our entities except patient only have three attributes: id, name, and type. Our patient object on the other hand has a lot more attributes beyond id and name:

Set fractions: Number of fractions a patient will be receiving in linac

Fractions slots: Specifies length of the fractions

Assigned machines: States which machines and doctor a patient is scheduled to

Priority: The priority given to the patient. Currently not used in any experiment.

Time entry/time leave: Timestamp for when the patient comes in/leaves the system

Received fractions: Counter for the number of fractions a patient has received.

4.8 Booking Process

A graph is presented in Figure 4.4. The graph describes how the booking process is constructed and how different parts of the simulation interacts with each other. From the Figure it is shown how the algorithm can be easily switched out.

4.9 Workflow

A workflow is defined as the path a patient takes through the treatment process. Since each cancer is different there is also a lot of workflows. In order to study the system only one workflow was chosen for all the experiments seen in Figure 4.5, although with the current model many more workflows can be constructed.

4.10 Verification

Due to the complex nature of the scheduling, verification with standard queuing theory methods proved complex and hard to implement, instead another approach was selected. Firstly a logical model was developed on paper with all the relations and different possible steps. Secondly the model was implemented(model used in all simulations) and tests were made to see that it performed correctly and produced reasonable results. This included checking throughput, patient system times, queuing length and schedule utilization. A third step was made by doing test runs with detailed print outs to see that the logic was correctly implemented. It was not possible to check all printouts so a couple of samples were selected.

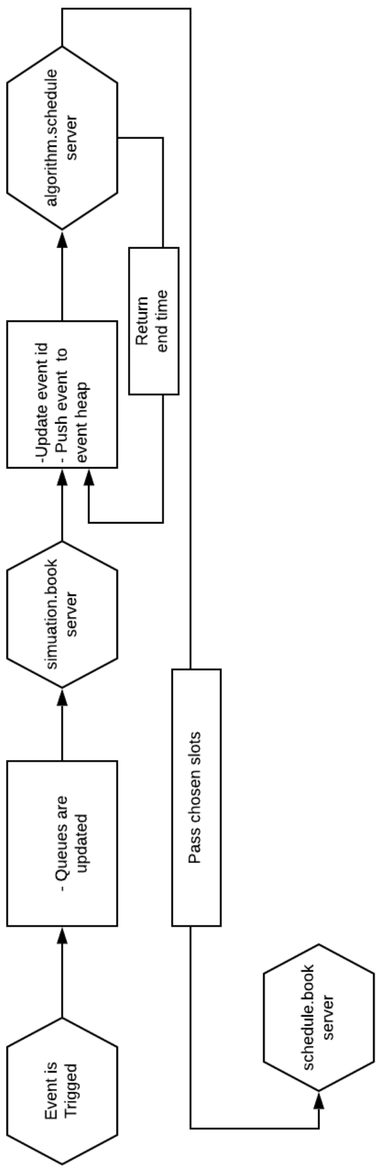

Figure 4.4: Booking Process

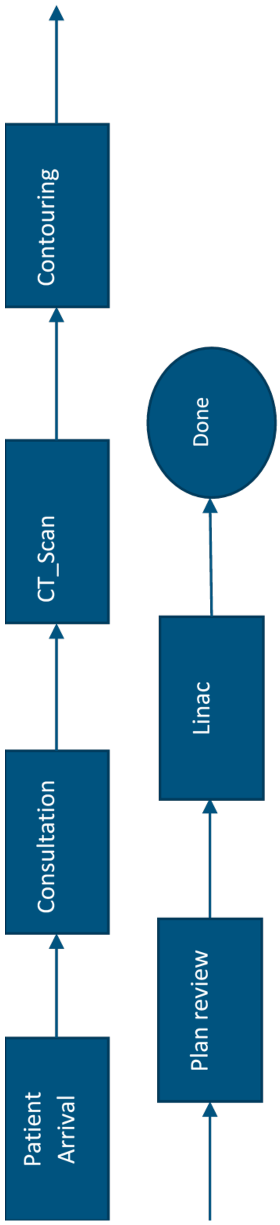

Figure 4.5: Workflow 1 for a simulation

Chapter 5

Experiments

In this chapter the setups of all simulations are detailed. Additionally, the purpose of each experiment is explained and also what is to be investigated.Previous studies do not mention how weekends are handled. Thus, in all experiments it is stated at which time a certain experiment can take place. No days are compressed and the simulations are run according to a real schedule based on the years 2020, 2021 and 2022. Furthermore, no holidays are taken into account in the schedules. To make the graphs more intuitive mean interrarival rate is instead presented as average patients arriving per week(PAR) according to: $Patient\ arrival/week = 7 * 1440/mean\ interarrival\ rate$

> **Experiment 1:** Bottleneck investigation of linac and doctors. Observations and measurement of how PST varies with PAR are also done. Same server is not active here.
>
> **Experiment 2:** Different schedules are implemented to see if schedule alignment has an effect on PST.
>
> **Experiment 3:** A load study of the system is made. In total there are two setups one with disturbance active and one without.
>
> **Experiment 4:** All disturbances are investigated separately with varying intensities. PAR is kept constant in all simulations.

5.1 Measurement setup

In many of our experiments studies of PST are made close to maximum capacity, which creates a problem because PST is only registered when a patient leaves which is not the case in a system with infinite queue growth. As a counteraction only patients arriving between days 200-385 have their PST registered for measurement, in total the simulation ran for 695 days. During the course of these experiments it was found that all system setups had a warmup period. Therefore measurements are taken after day 200. The simulation was run until all patients that entered between days 200-385 left.

5.2 Workflow

Although there are many possibilities to specify workflow with the current model only one was chosen for all experiments. The purpose of this is to make comparison between experiments and setups easier. Furthermore, it kept the complexity down making it easy to investigate fundamental properties of the system. The workflow that was chosen was based on previous studies and standard workflows used within RayCare. The workflow is specified in figure 4.5.

5.3 General settings

General settings common to all experiments were as follows:

- **Activities** - Each patient goes thorough all activities once except for the linac where each patient has a total of 30 fractions. Only one fraction is allowed per day

- **Activity length** - A consultation takes 60 minutes, whereas all other activities except linac take 30 minutes. The linac activity length is uniformly distributed between 10, 20 and 30 minutes.

- **Constraints** - On all activities there are booking constraints. A constraint tells how long a patient is allowed to wait before it is scheduled. Since overloading of the system is studied all booking constraints are relaxed to 90 days.If a constraint is violated the simulation shuts down.

5.4 Experiment 1

In this experiment an investigation is made into what happens when you set different servers as bottlenecks. The servers studied in this experiment are linacs and doctors. A total of 20 simulations are carried out for each setup. The only variable that changes between simulations in a setup is the PAR.

Although our main goal is to study bottleneck effects we will also look into PST, schedule utilization and variance.

5.4.1 Hypothesis

In the first setup doctors constitute the bottleneck. Therefore infinite queue growth should be seen for one of the activities in their schedule when PAR increases.In the second setup it should be the other way around and instead the linac queue grows towards infinity.

5.4.2 Setting that are the same in both simulations

Arrivals

Event	Arrival process	Arriving on weekend	Patient treated by same server
Patient arrival	Poisson process	Yes	No

Table 5.1: Arrival settings - Experiment 1

Servers

Server	Quantity	Schedule	Slots	Schedule algorithm
Doctor	2	08:00 - 20:00	Json-file	Greedy
CT scan	1	08:00 -17:00	All	Greedy
Linac	3/6	08:00 - 20:00	All	Greedy

Table 5.2: Servers - Experiment 1

5.4.3 Schedules

In setup 1 the doctor serve as the bottleneck and therefore there are less slots. In setup 2 the linac serve as the bottleneck and thus more slots are allocated for each activity in the doctors schedule.

Setup	Doctor	Consultation	Contouring	Plan review
1	1	Monday: 8:00-12:00 Tuesday 8:00-12:00	Thursday: 8:00-12:00	Thursday: 13:00-17:00
1	2	Monday: 8:00-12:00 Tuesday 8:00-12:00	Thursday: 8:00-12:00	Thursday 13:00-17:00
2	1	Monday: 8:00-14:00 Tuesday 8:00-12:00	Thursday: 8:00-13:00	Thursday: 13:00-18:00
2	2	Monday: 8:00-14:00 Tuesday 8:00-12:00	Thursday: 8:00-13:00	Thursday: 13:00-18:00

Table 5.3: Schedule - Experiment 1

5.4.4 Setup 1

In this setup an investigation is made into what happens when we set doctors as the bottleneck with a maximum booking delay of 90 days. Here we increase the number of linacs to 6 in order to ensure that the bottleneck will be the doctors. Mean interarrival time is the average time between two patient arrivals.

Mean interarrival time (min)	800	750	700	650	610

Table 5.4: PAR - Setup 1 - Experiment 1

5.4.5 Setup 2

In this setup an investigation is made into what happens when we set the linac as a bottleneck with a maximum booking delay of 90. The number of linacs is decreased to 3 in order to ensure that the bottleneck will be the linacs. Doctors capacity is also increased 30 percent compared to the settings in setup 1 to ensure that linacs will constitute a bottleneck.

Mean interarrival time (min)	900	850	800	775	750

Table 5.5: PAR - Setup 2 - Experiment 1

5.5 Experiment 2

In this experiment the effect of rearranging the doctors schedules is studied. Capacity and hence number of slots are kept constant between all simulations. The schedule is chosen in such a way that the theoretical minimal treatment is different between all schedules. Although one study [4] has been made on the effect of aligning doctors schedules it did not mention in depth how much greater PST was. Furthermore, this experiment is an investigation made into both mean patient system time (MST) and mean of 10% greatest system times (MGST10) which was not done in the previous study.

5.5.1 Hypothesis

The previous study indicated that minimal patient system time has an effect on PST overall. Therefore it is likely to see this effect in this experiment. Average schedule utilization and variance should not change since on average the same number of patients are treated each week and capacity is kept constant.

5.5.2 Setting that are the same in all simulations

Simulation length

Simulation length is 695 days.

Arrivals

Settings are given in table 5.1. Mean interarrival time is : 750 min or

Servers

Settings are given in table 5.2.

5.5.3 Schedule

Simulation	Doctor	Consultation	Contouring	Plan review
1	1	Monday: 8:00-12:00 Tuesday 8:00-12:00	Tuesday: 13:00-17:00	Wednesday: 14:00-18:00
1	2	Wednesday: 8:00-12:00 Thursday 8:00-12:00	Thursday: 13:00-17:00	Friday: 14:00-18:00
2	1	Monday: 8:00-12:00 Tuesday 8:00-12:00	Tuesday: 13:00-17:00	Wednesday: 14:00-18:00
2	2	Monday: 8:00-12:00 Tuesday 8:00-12:00	Wednesday: 13:00-17:00	Thursday: 14:00-18:00
3	1	Monday: 8:00-12:00 Tuesday 8:00-12:00	Thursday: 13:00-17:00	Wednesday: 14:00-18:00
3	2	Monday: 8:00-12:00 Tuesday 8:00-12:00	Thursday: 13:00-17:00	Tuesday: 14:00-18:00

Table 5.6: Schedule - Experiment 2

5.6 Experiment 3

An investigation is made into how PST is affected by increased PAR. Two setups are constructed one with disturbances active and one without. Furthermore, variance and mean of the doctors schedule are also investigated. Although the setup without disturbance is similar to Experiment 1 it has one major difference and that is: Each patient is served by the same server throughout the workflow. The main focus in this experiment is the doctors schedule. The purpose of this experiment is to see how sensitive the system is to changes in patient arrival rates at high schedule utilization, lower utilization $< 60\%$ is not interesting to study, because most clinics want to have as high utilization as possible.

5.6.1 Hypothesis

Having the same server should contribute to increased PST compared to Experiment 1 and schedule utilization should be lower, mainly because any server cannot treat all patients which on average makes the system less effective. Furthermore, disturbances should decrease schedule utilization especially when a cancellation event is active. The maximum PST without an overloaded system should not differ significantly between both setups, primarily because there is no difference between the setups except the disturbance.

5.6.2 Setting that are the same in both setups

Simulation length

Simulation length is 695 days.

Mean interarrival time (min)	1000	800	750	700	650	600	550	540	530	520	510

Table 5.7: PAR - Experiment 3

Event	Arrival process	Arriving on weekend	Patient treated by same server
Patient arrival	Poisson process	No	Yes

Table 5.8: Arrival settings

Arrivals

Servers

Server	Quantity	Schedule	Slots	Schedule algorithm
Doctor	2	08:00 - 20:00	Json-file	Greedy
CT scan	1	08:00 -17:00	All	Greedy
Linac	5	08:00 - 20:00	All	Greedy

Table 5.9: Servers - Experiment 3

Schedule

Simulation	Doctor	Consultation	Contouring	Plan review
1	1	Monday: 8:00-14:00 Tuesday 8:00-12:00	Tuesday: 13:00-18:00	Friday: 14:00-19:00
1	2	Thursday: 8:00-14:00 Friday 8:00-12:00	Thursday: 8:00-13:00	Friday: 14:00-19:00

Table 5.10: Schedule - Experiment 3

5.6.3 Setup 1

Setup 1 contains all simulations run without any disturbances. There is one simulation for each inter arrival rate as specified in table 5.7

5.6.4 Setup 2

Setup 2 contains all simulations run with disturbances. In this case all disturbances are active. The parameters for each disturbance are the same throughout all simulations. There is one simulation for each interarrival rate as specified in Table 5.7

Disturbances

Disturbances are given in table 5.11

Event	Event process	Mean interrarrival time	Only weekdays
Patient rebook	Poisson	8000 min	No
Linac rebook	Poisson	2000 min	No
Doctor rebook	Poisson	10000 min	No
Sick doctor	Poisson	20000 min	Yes

Table 5.11: Disturbances - Experiment 3

5.7 Experiment 4

In this experiment a study is made into how sensitive the system is to different disturbances. Each disturbance is investigated separately. For all disturbances PST is investigated. Furthermore, schedule utilization and variance are also analyzed. Choice of disturbance intensity interval is set with respect to the number of servers corresponding to each disturbance. Different PAR were chosen in order to study the system as close to max capacity as possible.

5.7.1 Hypothesis

Cancellation disturbances should have the greatest effect on the system since on average they decrease the number of available slots. It should mainly affect PST times and utilization. Rebooking disturbances should have a lesser effect but it should still be noticeable since it disturbs the process and therefore schedule variance should increase.

5.7.2 Setting that are the same in both setups

Simulation length

Simulation length is 1000000 minutes.

Arrivals

Setup	Event	Arrival process	Mean interrarrival time	Arriving on weekend	Patient treated by same server
Rebook linac	Patient arrival	Poisson process	560min	Yes	Yes
Rebook patient	Patient arrival	Poisson process	560min	Yes	Yes
Rebook doctor	Patient arrival	Poisson process	560min	Yes	Yes
Sick doctor	Patient arrival	Poisson process	650min	No	Yes

Table 5.12: Disturbance interrarival rates - Experiment 4

Servers

The servers are the same for all setups.

Server	Quantity	Schedule	Slots	Schedule algorithm
Doctor	2	08:00 - 20:00	Json-file	Greedy
CT scan	1	08:00 -17:00	All	Greedy
Linac	5	08:00 - 17:00	All	Greedy

Table 5.13: Servers - Experiment 4

Schedule

The schedule is given in figure 5.10.

5.7.3 Sick doctor

In this setup the effect of having a sick doctor is studied. Disturbance intensity for each simulation is varied according to table 5.14. The only disturbance that is active in this is sick doctor.

Arrival on weekends: No

Event process: Poisson

Mean interarrival time (min)	30000	20000	15000	12500	10000	7500	5000

Table 5.14: Disturbance sick doctor - Experiment 4

5.7.4 Patient rebook

In this setup the effect of having a patient re-booking his/her appointment with a doctor is studied. Disturbance intensity is varied according to table 5.15. The only disturbance that is active in this setup is patient rebook.

Arrival on weekends: Yes

Event process: Poisson

Mean interarrival time (min)	10000	9000	8000	7000	6000	5000	4000

Table 5.15: Disturbance patient rebok - Experiment 4

5.7.5 Linac rebook

In this setup the effect of having a patient re-booking his/her scheduled Linac machine slot is studied. Disturbance intensity is varied according to table 5.16. The only disturbance that is active in this setup is linac rebook.

Arrival on weekends: Yes

Event process: Poisson

Mean interarrival time (min)	8000	7000	6000	5000	4000	3000	2000

Table 5.16: Disturbance linac rebook - Experiment 4

5.7.6 Doctor rebook

In this setup the effect of having a doctor re-booking his/her appointment is studied. Disturbance intensity is varied according to table 5.17. The only disturbance that is active in this setup is doctor rebook.

Arrival on weekends: Yes

Event process: Poisson

Mean interarrival time (min)	12000	10000	9000	8000	7000	6000	5000	4000

Table 5.17: Disturbance doctor rebook - Experiment 4

Chapter 6

Results and Discussion

In this chapter all the results are presented. In all graphs the vertical bars marks the 95 % confidence intervals for each measurement.

6.1 Experiment 1

6.1.1 Results

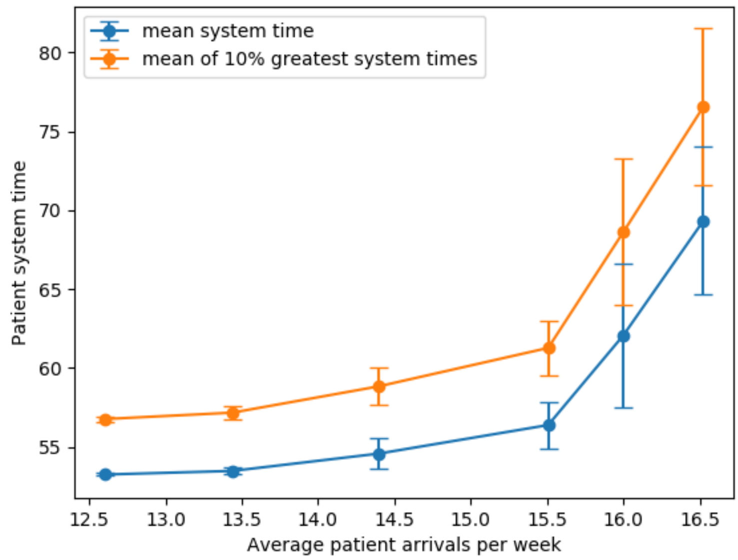

Figure 6.1: PST - Setup 1 - Experiment 1

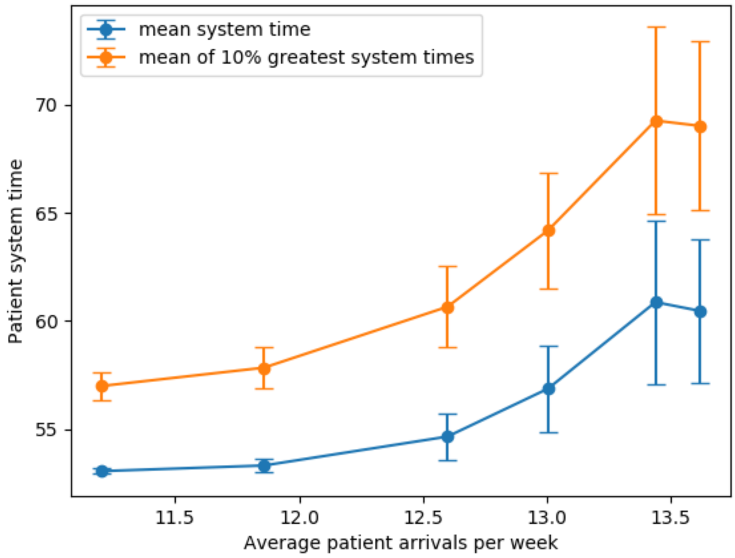

Figure 6.2: PST - Setup 2 - Experiment 1

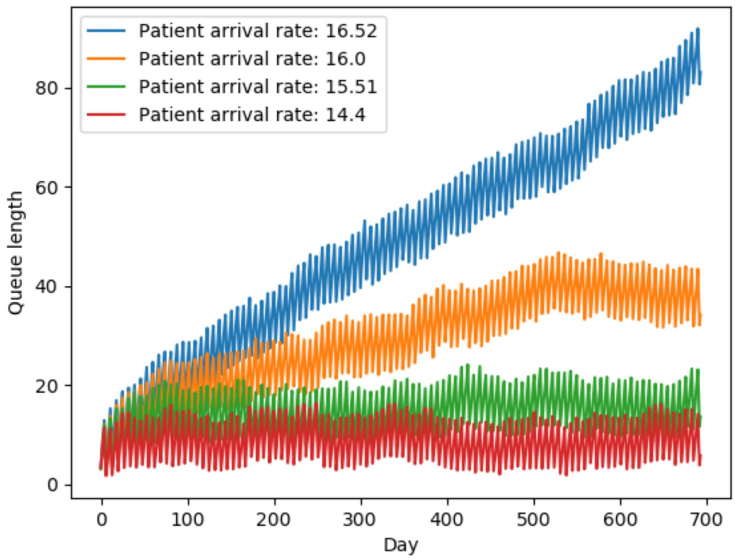

Figure 6.3: Queue consultation Setup 1 - Experiment 1

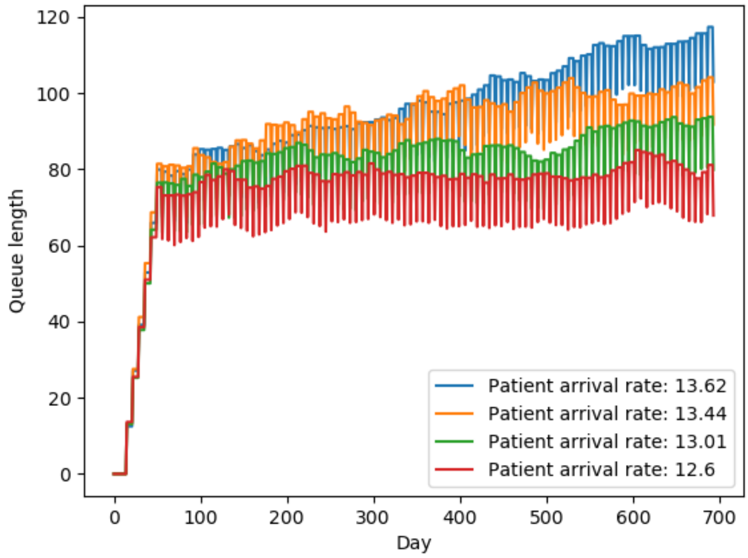

Figure 6.4: Queue Linac- Setup 2 - Experiment 1

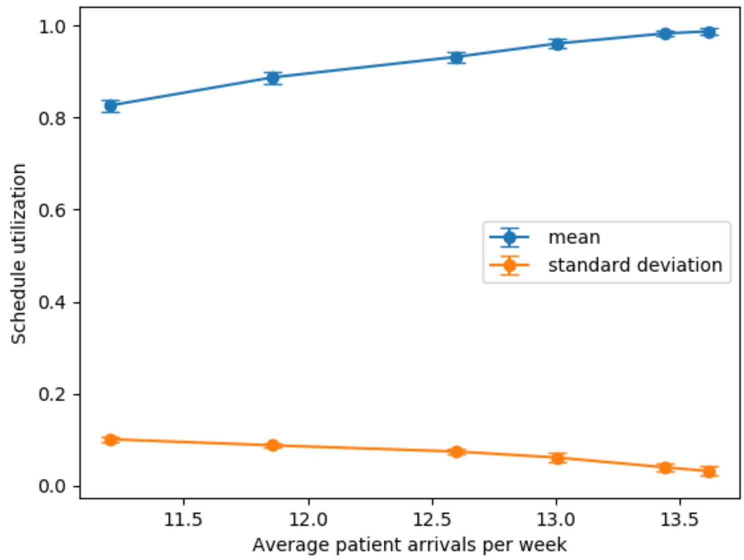

Figure 6.5: Linac schedule utilization - Setup 2 - Experiment 1

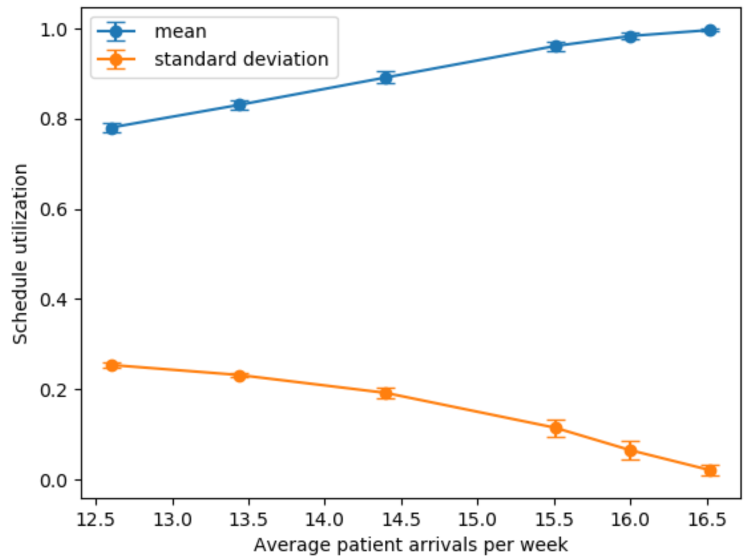

Figure 6.6: Doctors schedule utilization - Setup 1 - Experiment 1

6.1.2 Discussion

Figure 6.3 shows that in setup 1 doctors constitute the bottleneck. This can be seen because the consultation queue is growing towards infinity. The system becomes overloaded when PAR $>$ 16. All activities in the doctors schedule have the same capacity for handling patients each week. Consequently, it might be strange that consultation is the bottleneck, although this might depend upon that it is the first station in our workflow or that arrival intensity fluctuates more. In figure 6.4 the bottleneck is caused by the linac, because the Linac queue grows towards infinity when PAR $>$ 13.5

Although there are some overlapping in the confidence interval in figure 6.1 and 6.2 the non linear effect is still present for the stable system, especially for MGST10. Only setup 1 seem to have a threshold value for PAR when the PST starts to increase drastically after PAR $>$ 15.5. One reason why this might be is that if on any given day each activity always has a slot available directly an increase in PAR will not affect PST at all, since the system can perform the activity without delay. But after a certain PAR is reached this might no longer be possible because the system starts to become congested and then each activity is delayed, which leads to a drastic increase in PST. The threshold is maybe easy to predict in smaller systems but will probably be impossible to do analytically in larger systems. Setup 2 (linac bottleneck) does not seem to have any threshold present, it does instead seem to increase non linearly for all values measured. This might depend on that the threshold values happen at a lower PAR for the linac or not at all.

Figure 6.5 and 6.6 show that achievable utilization in both cases is close to

100% without the system getting overloaded. Finally, variance is steadily decreasing with increasing PAR.

Conclusions

From the figures it is clear that our hypothesis was correct about which server would be the bottleneck in each setup.

A conclusion to be drawn from this experiment is that regardless of which server constitutes the bottleneck PST tends to increase non linearly. Furthermore, this effect seems to be stronger for MGST10. In both setups it was possible to reach almost 100 % utilization. Even though schedules have the same number of slots for different activities it does not mean that all activities become bottlenecks such as in the case with the consultation activity. One can also note from the wider confidence intervals that PST starts to vary more significantly when the system becomes congested.

6.2 Experiment 2

6.2.1 Results

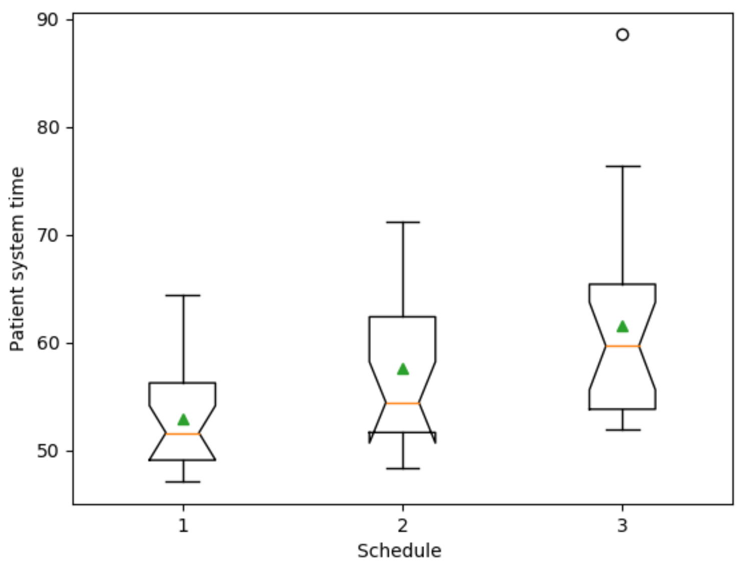

Figure 6.7: MST - Doctor schedule - Experiment 2

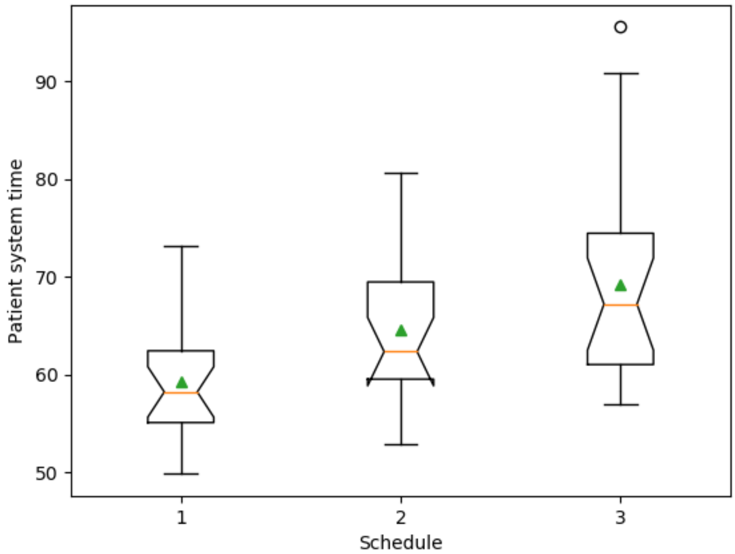

Figure 6.8: MGST10 - Doctor schedule - Experiment 2

6.2.2 Discussion

Figure 6.7 shows that there is an overlap between the schedules. But one can clearly see that there is a significant difference between schedule 1 and schedule 3 with the notches only having a small overlap, indicating that slot rearrangement not only affects minimum PST but also MST. Figure 6.7 with MGST10 also exhibits this effect with it being more significant between schedule 1 and schedule 3. Therefore, it is hard to draw any conclusion for schedule 2. Utilization between the different schedules does vary some and even though the confidence intervals are not overlapping for all measurements the difference is so small that one can conclude that slot alignment only has a marginal effect on variance and mean utilization.

6.2.3 Conclusion

Poorly aligned schedules cause a significant change in PST. It does not only change minimal treatment time it also changes MST and MGST10. Therefore it is important when scheduling servers to take alignment of activities into account. This can be done as stated in previous studies either manually or with mixed integer linear programming. Schedule utilization and variance do not seem to change as long as the number of slots is kept constant.

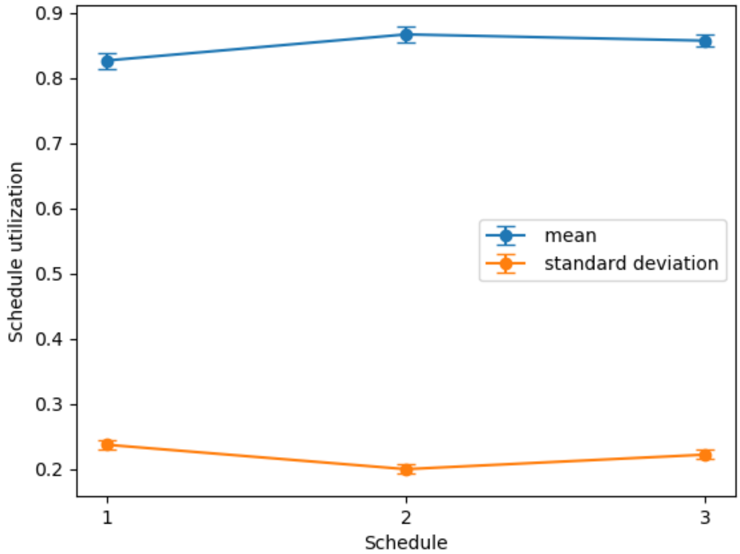

Figure 6.9: Utilization and variance Experiment 2

6.3 Experiment 3

In this experiment the discussion is mainly focused around comparing results internally. But since experiment 1 - setup 1 is similar some comparison are made with it.

6.3.1 Results

Results are presented below. Figure for patient system time, queue length for doctors consultation and doctors schedule utilization are presented.

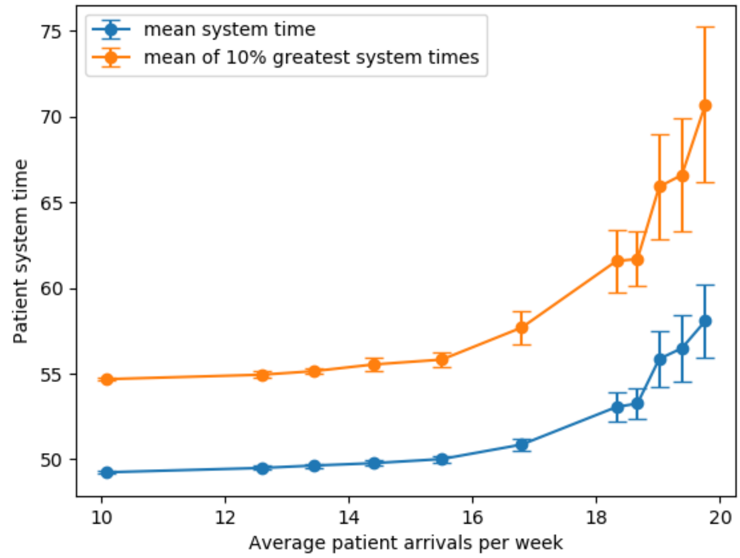

Figure 6.10: PST - No disturbance - Experiment 3

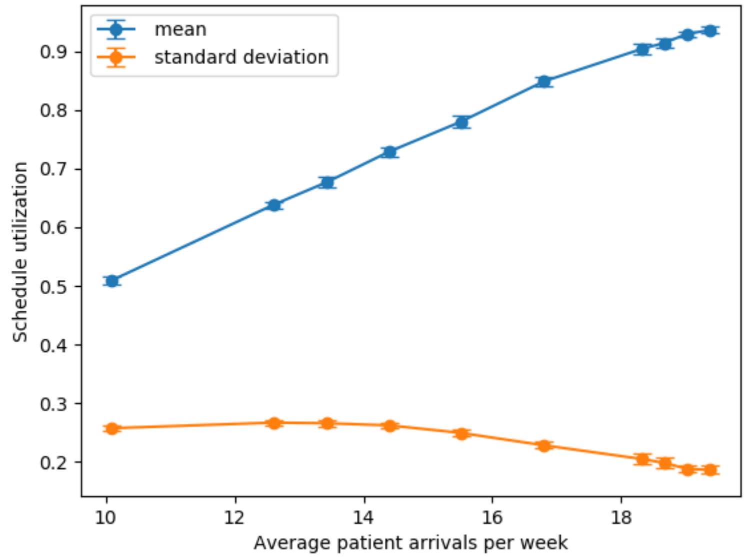

Figure 6.11: Doctors schedule utilization - No disturbance - Experiment 3

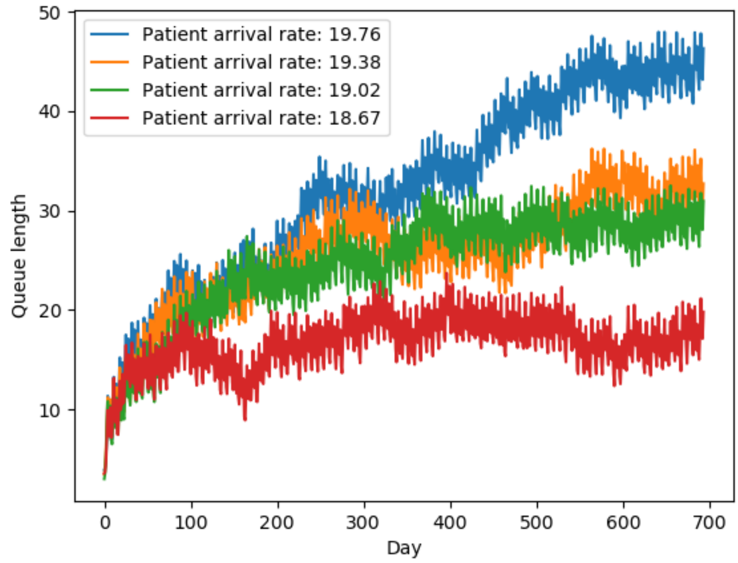

Figure 6.12: Consultation queue - No disturbance - Experiment 3

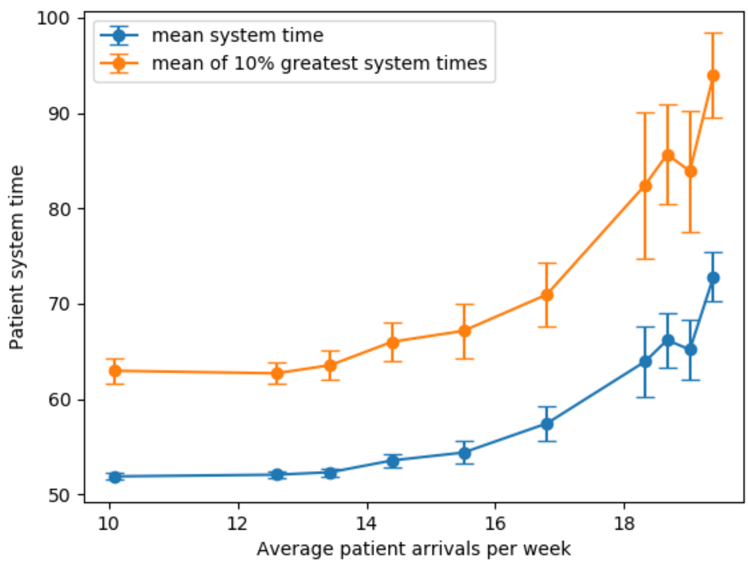

Figure 6.13: PST - Disturbance - Experiment 3

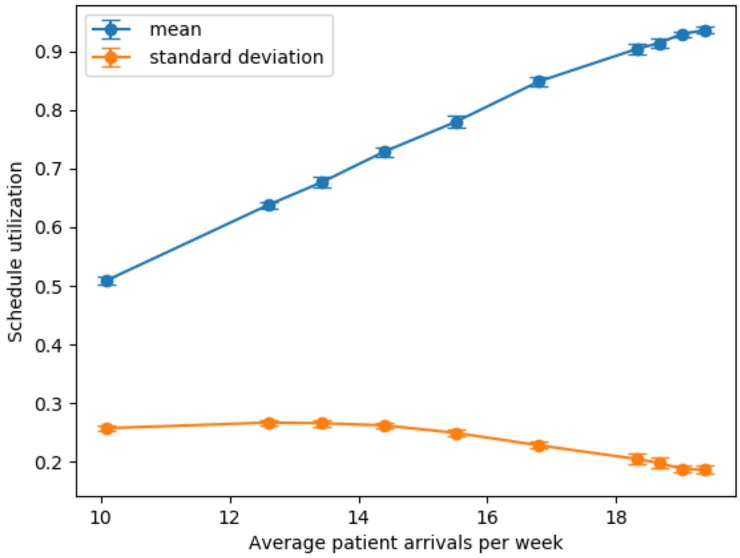

Figure 6.14: Doctors schedule utilization - Disturbance - Experiment 3

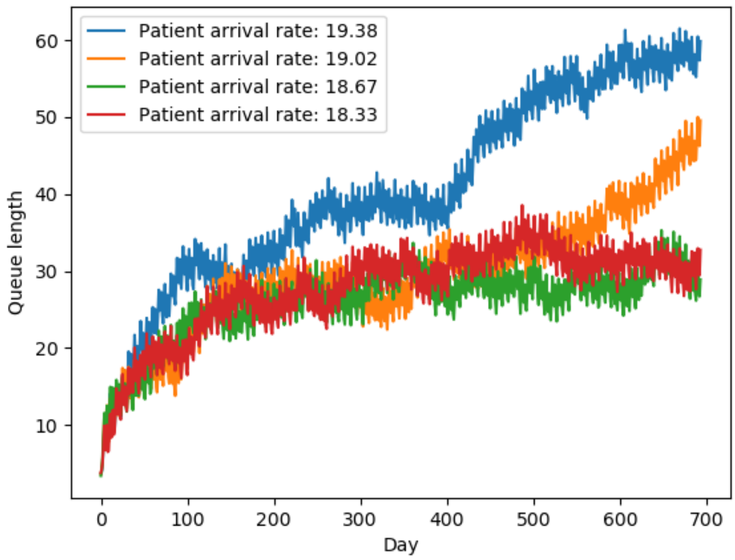

Figure 6.15: Consultation queue - Disturbance - Experiment 3

6.3.2 Discussion

The results showed that consultation activity in the doctors schedule caused the bottleneck. In this experiment all activities in the doctors schedule have equal capacity. The reason why consultation activity constitutes the bottleneck is the same as in experiment 1.

Both figure 6.10 and 6.14 show a non linear relation for PST as PAR increases. The non linearity seems to be especially strong for MGST10. Confidence intervals are wide and overlapping to some extent but even with them accounted for the non-linear effect is still clear. Both setups seem to have a threshold where the PST starts to increase; below that threshold there seems to be only a small increase in PST. Without disturbance it is around PAR > 16 and with disturbance PAR > 14. Interestingly, when the disturbance was active, MST and MGST10 were greater than without even when the system was not overloaded. Therefore the disturbances introduce some kind of delay to the system, because otherwise the PST should be the same for both setups when maximum capacity is reached. This is supported by looking at PST for low PAR values, where the MGST10 is significantly higher when the disturbance is active. Conclusively, disturbances introduce a delay that seem to have a constant contribution but it might also scale with increasing PAR.

Since there are different PAR thresholds for each setup, PST starts to increase at different mean utilization in the respective setups. Figures 6.14 and 6.11 show that the utilization threshold is around 72% and 80 % respectively. With disturbance activated the maximum utilization that is achievable without infinite queue growth is around 90 % and 95% without disturbance . Therefore, a combination of active disturbances decreases the maximum achievable utilization. Rebooking events should in theory not decrease schedule utilization since an old slot is made available for a new booking. There could however be an effect if a rebooking is performed close to the current time and the old slot is not booked again. Cancellation events on the other hand cancel slot and should therefore have a direct effect on the achievable utilization.

Comparison and similarities with experiment 1

Throughout both experiment 1 and 3 there has been a difference between MST and MGST10. Since a person needs to go through all activities there is a theoretical minimum PST time. The minimum PST depends upon which day of the week a patient enters the system, therefore it is reasonable to have a difference between MGST10 and MST even though the load on the system is low.

What can be noticed it that for low PAR both MGST10 and MST are greater in experiment 1, one reason for this is that the experiments have different schedules, as can be seen in experiment 2. Reasonably the same server constraint should increase PST since the schedule on average cannot be utilized to the same degree but in this case the schedule effect probably is stronger. The maximum schedule utilization for a stable system decreased from 98% in experiment 1 figure 6.1 to 95% in experiment 3 with no overlap in confidence interval, indicating that schedule utilization decreases with the same server active. To be certain one should run more test with the same schedules.

Conclusion

From both setups it is conclusive that there is a nonlinear increase in PST as PAR increases. Furthermore, the system has a PAR threshold value/range where it is insensitive to varying PAR below and sensitive above just as in experiment 1. This value varied depending on if disturbances were active or not. Both experiment 1 and 3 had decreasing variance with increased PAR with non overlapping confidence intervals, so it is therefore conclusive that when the load increases on the system schedule variance decreases. Finally, disturbances greatly affect both PST and utilization negatively although it is not conclusive in this experiment exactly which disturbance or combination.

Our hypothesis was confirmed except that disturbances did contribute to longer PST for a non overloaded system.

6.4 Experiment 4

6.4.1 Results

The mean interarrival times was chosen differently for each disturbance, mainly because the magnitude of each disturbance was different per intensity. Disturbance intensities were chosen in relation to the number of patients that come to the clinic each week.

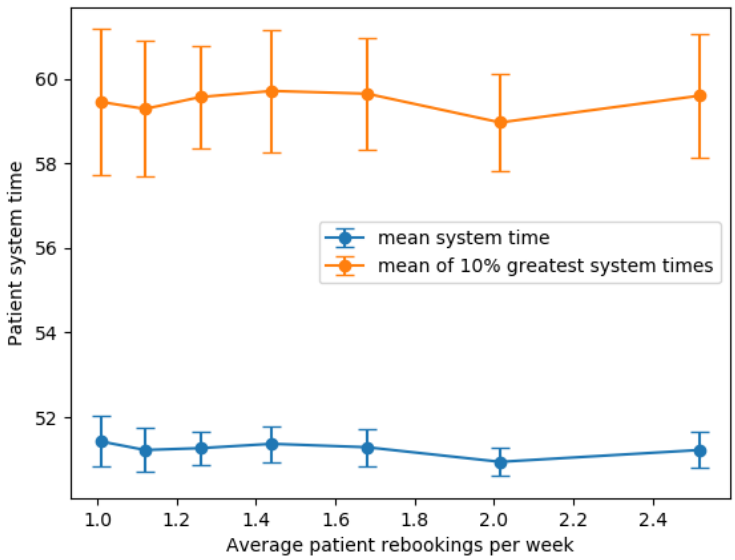

Figure 6.16: PST - Patient rebook - Experiment 4

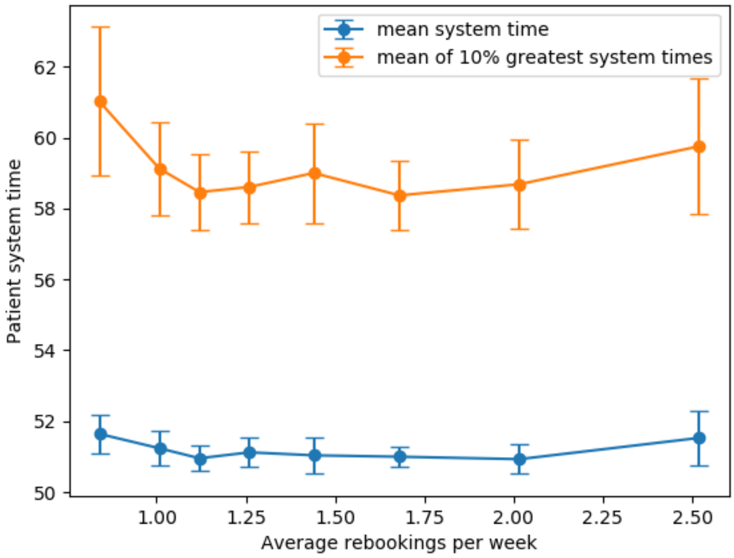

Figure 6.17: PST - Doctor rebook - Experiment 4

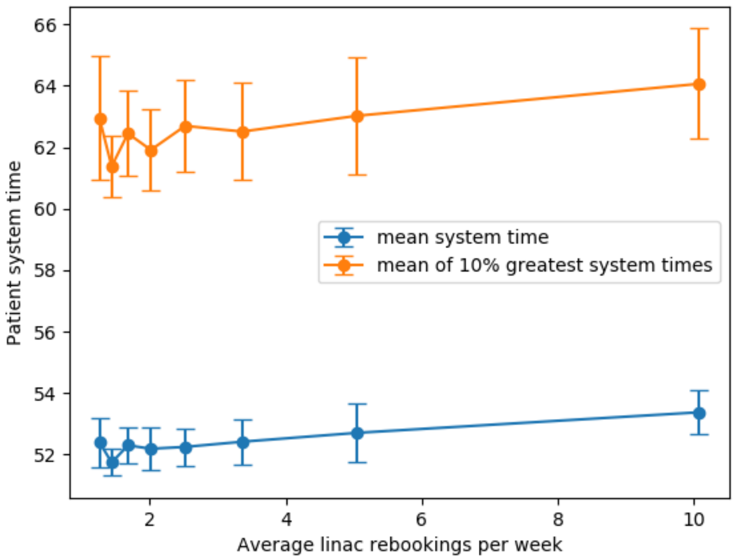

Figure 6.18: PST - Linac rebook - Experiment 4

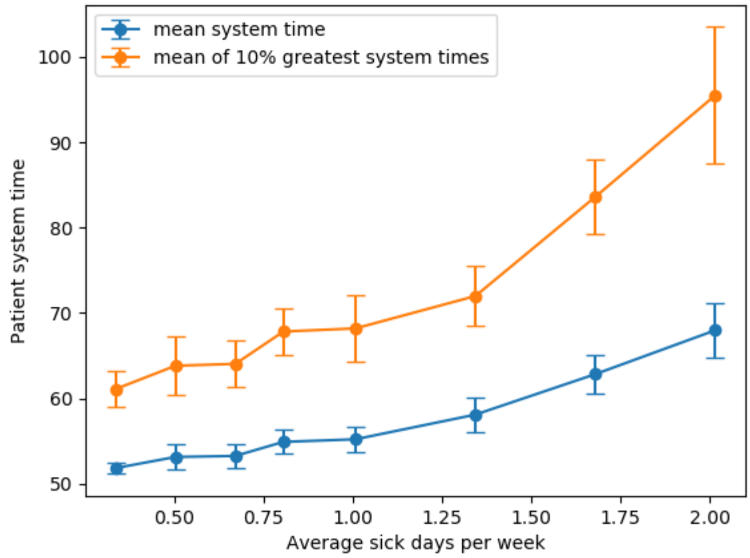

Figure 6.19: PST - Sick doctor - Experiment 4

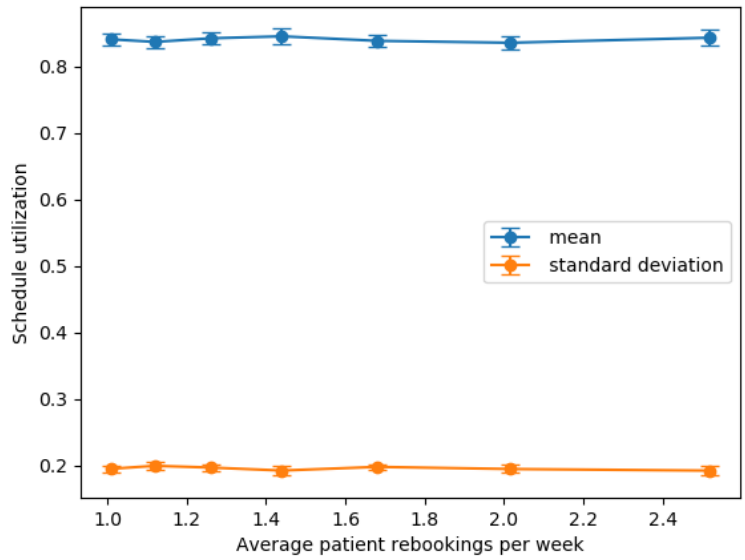

Figure 6.20: Patient Rebook Utilization - Experiment 4

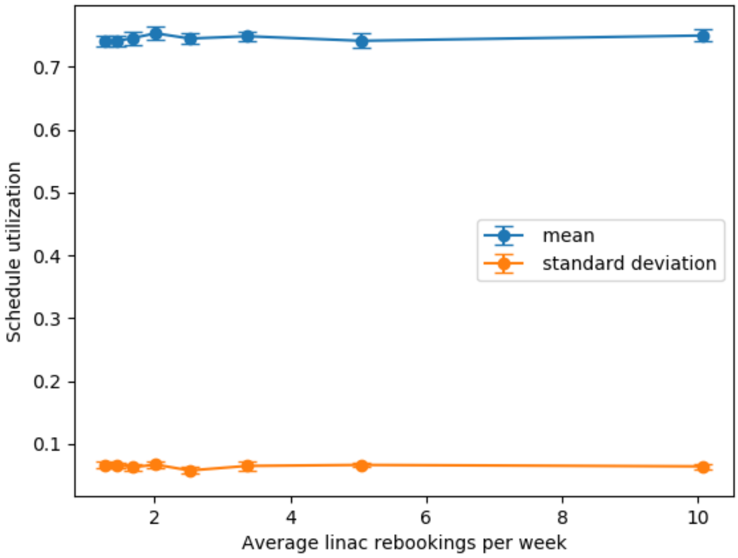

Figure 6.21: Linac Rebook - Utilization - Experiment 4

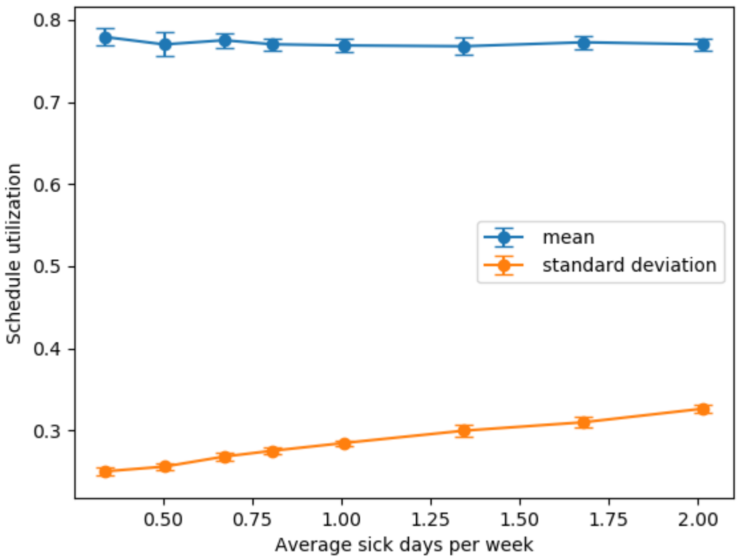

Figure 6.22: Sick doctor - Utilization - Experiment 4

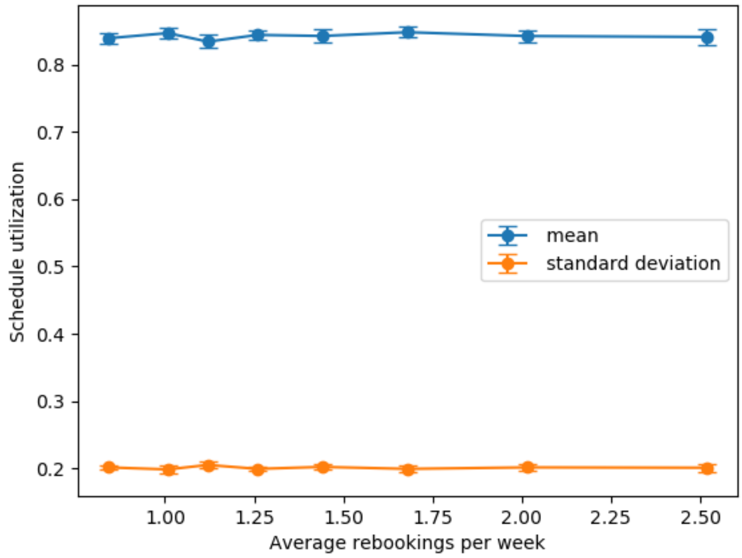

Figure 6.23: Doctor rebook - Utilization - Experiment 4

6.4.2 Discussion

PST for all rebooking events in figures 6.17, 6.16 and 6.18 all have overlapping confidence intervals for almost all measurements for both MST and MGST10. Thus one can not conclude that rebooking events affect PTS. This implies that all slots that are made available when a patient is rebooked will be booked by another patient and therefore the PST is not affected. Variance and utilization for rebooking events in figures 6.20, 6.21 and 6.23 have overlapping confidence intervals for almost all measurements, indicating that rebooking disturbances do not affect schedule utilization, confirming our hypothesis. It might be strange that rebooking does not change utilization variance, but utilization is measured after all events have taken place each day at 23.59 and when a person is rebooked it happens before the measurement is taken. Therefore, if the old slot is booked the rebooking event will not contribute to increased variance or a change in utilization.

In Figure 6.19 it is clear however that there is a relation between disturbance intensity and PST. It is not entirely clear whether or not the relation is non linear, but it seems more non-linear. A major difference between rebooking events and sick doctor, is that when a doctor is sick the number of available slots decreases which is not the case otherwise. Interestingly, schedule utilization stays constant as seen in Figure 6.22. When a doctor is sick the patients are booked on another day, and the old slots are cancelled. Furthermore, in this experiment PAR stays constant which indicates that even though some slots are cancelled the schedule utilization will stay constant, since on average the same number of patients are served each week as long as the system does not get overloaded. As average doctor sick days per week increase, utilization variance also increases. It can be explained by the fact that for a cancellation disturbance the old slot cannot be rebooked which will be registered when utilization is measured which is the opposite of a rebooking event.

Conclusion

Conclusively rebooking disturbances have little to no effect on PST, utilization and variance. The main contribution is the rebooking mechanism explained in the discussion. Cancellation events such as sick doctor have an effect on the system and PST seems to increase non linearly. Finally, utilization stays the same but as disturbance intensity increases so does the variance.

Chapter 7

Conclusions

7.1 Bottlenecks

In experiment 1 it was concluded that any server can serve as the bottleneck depending on how the clinic is constructed. In all other experiments where overloading of the system was studied, doctors constituted the bottleneck, more specifically consultation. This was the case even though all doctor activities had equal capacity. The cause was not further investigated, but likely due to the fact that consultation was first in the workflow.

7.2 Utilization And Variance

One of the main purposes of this thesis was to investigate schedule utilization properties under various settings. What has been found to be conclusive throughout all experiments is that increased system load decreases utilization variance for all servers. The main cause is that as PAR grows, average queue length to each server also grow which leads to lower variance.
Experiment 4 showed that rebooking disturbances had no effect on either utilization variance or mean utilization. One of the main causes was that almost always the old slot is booked again and thus not changing the utilization on the day that rebooking occurred and thus not increasing the variance. Cancellation disturbance increased variance as disturbance intensity increased. The likely reason for this is that when an event activity is cancelled it affects the utilization on that particular day. Finally, disturbances also affected maximum achievable utilization, through experiment 4 it was found that likely only sick doctor/cancellation event are the cause.

7.3 Patient System Times

It was found throughout all experiments that PST increased non linearly as a function of PAR. There seems to be a threshold for when the system starts to become sensitive to fluctuations in PAR. This was especially clear when the doctors constituted the bottleneck in experiment 1 and 3. It was interpreted as the system congestion point. Below the threshold the system has excess capacity and

therefore can take additional patients without increasing PST and above it starts to increase drastically. Furthermore, it was found that when disturbances were active, PST increased for both low and high doctor schedule utilization.
Through experiment 4 there is a strong indication that the only disturbances contributing to increased PST is sick doctor. This is because all the other disturbances were rebooking and not cancellation disturbances. Thus, it was concluded that the only likely contributing disturbance in experiment 3 was sick doctor.
As in previous studies it was found that schedule alignment affect treatment times. It did not only affect minimal treatment time but also MST and MGST10.

7.4 Future Work

In this thesis only a small workflow was constructed. Normally clinics are much more complex and have a number of workflows, patient priorities, stations and resources. Therefore, a future study should expand the implemented model so that it more closely resembles a real clinic. This will also open up the possibility to validate the model with real world data. Furthermore, our model did only contain doctors so it would be wise to expand the resources to make them more specific as in a real world clinic. Arrival time and schedules were static for each simulation, which may not be the case in a real clinic and should therefore also be implemented. Finally one of the main goals of the company is to use this thesis to make clinic more effective, therefore development and investigation of the algorithm should be made.

References

[1] ALlen, T. T. *Introduction to Discrete Event Simulation and Agent-based Modeling*. Springer, Reading, Massachusetts, 2011.

[2] Babshov, V., Aivas, I., Begen, M., Rodrigues, G., Souza, D., and Lock, M. A pattern-based approach of radiotherapy scheduling. *Clinical Oncology 298* (2017), 385–391.

[3] Banks, J., Carson, J. S., Nelson, B., and Nicol, D. *Discrete-Event System Simulation*. Pearson Education Limited, London, England, 2013.

[4] Bikker, I. A., Kortbeek, N., OS, R. M., and Boucherie, R. J. Reducing access times for radiation treatments by aligning the doctors schemes. *Operations research for Health Care 7* (2015), 111–121.

[5] Conforti, D., Guerriero, F., and Guido, R. Non-block scheduling with priority for radiotherapy treatments. *European Journal of Operational Reserach 201*, 1 (2010), 289–296.

[6] Degerfält, J., Moegelin, I.-M., and Sharp, L. *Radiation Therapy(Strålbehandling)*. Studentlitteratur, Stockholm, Sweden, 2008.

[7] Fishman, G. S. *Principles of Discrete Event Simulation*. John Wiley Sons, Inc., New York, USA, 1978.

[8] Joustra, P., Kolfin, R., van Dijk, N., Koning, C. C. E., and Bakker, P. Reduce fluctuations in capacity to improve the accessibility of radiotherapy treatment cost-effectivley. *Flexible Services and Manufacturing Journal volume 24* (2012), 448–464.

[9] Macron, E., Pommier, P., and Jacquemin, Y. A pattern-based approach of radiotherapy scheduling. *IFAC World Congress 201*, 1 (2010), 289–296.

[10] Petrovic, D., Morshed, M., and Petrovic, S. Genetic algorithm based scheduling of radiotherapy treatments for caner patients. *Artificial Intelligence in Medicine 5651* (2009), 155–189.

[11] Siqao, L., Geng, N., and Xiaolan, X. Meeting patient waiting time targets. *Robotics Automation Magazine 22*, 2 (2015), 51–63.

[12] Vieira, B., Demirtas, D., van de Kamer, J. B., Hans, E. W., and van Harten, W. Improving workflow control in radiotherapy using discrete-event simulation. *BMC Medical Informatics and Decision Making 19* (2019), 199–213.

[13] Werker, G., Saure, A., , French, J., and Shechter, S. The use of discrete-event simulation modelling to improve radiation. *Radiation Therapy and oncology 92* (2009), 76–82.