



OPTIMIZING COMPUTED
TOMOGRAPHY SCAN
SCHEDULING FOR
SUSTAINABLE ENERGY USAGE:
A SIMULATION STUDY AT
MEDISCH SPECTRUM TWENTE

Bachelor Thesis

ABSTRACT

In this research, we investigated sustainable cost-saving opportunities to reduce energy consumption of CT scanners in the radiology department of Medisch Spectrum Twente using a Monte Carlo simulation approach.

Schuppert, W.L.C. (Ward, Student B-IEM)

Supervisor 1: Amin Asadi (University of Twente)

Supervisor 2: Erwin Hans (University of Twente)

Supervisor 3: Mark van Kooten (Medisch Spectrum Twente)

Optimizing Computed Tomography Scan Scheduling for Sustainable Energy Usage: A Simulation Study at Medisch Spectrum Twente

BACHELOR THESIS

AUTHOR:

SCHUPPERT, W.L.C. (WARD, STUDENT B-IEM)

S2016877

25-MAY-23

THIS REPORT WAS WRITTEN AS PART OF THE BACHELOR THESIS MODULE OF THE INDUSTRIAL ENGINEERING AND
MANAGEMENT EDUCATIONAL PROGRAM.

UNIVERSITY OF TWENTE.



UNIVERSITY OF TWENTE

INDUSTRIAL ENGINEERING AND MANAGEMENT, PO BOX
217, 7500 AE ENSCHEDE, TEL. +31(0)53 489 9111

SUPERVISORS

DR. A. ASADI (UNIVERSITY OF TWENTE)
PROF. DR. IR. E. W. HANS (UNIVERSITY OF TWENTE)

MEDISCH SPECTRUM TWENTE

MEDISCHE TECHNIEK, KONINGSTRAAT 1, 7512 KZ
ENSCHDE, TEL. +31(0)53 487 2000

SUPERVISOR

M. VAN KOOTEN, MSC (MEDISCH SPECTRUM TWENTE)

MANAGEMENT SUMMARY

Problem definition

Medisch Spectrum Twente (MST) is a hospital in Enschede, the Netherlands. As the healthcare sector faces constant pressure to improve the level of care, MST has identified key objectives including sustainability, care quality, education, and scientific advancements (MST, 2022). In line with these goals, our research seeks to identify sustainable optimization options.

The Dutch government has set ambitious targets to reduce CO₂ emissions by 49% by 2030 compared to 1990 levels (Rijksoverheid, 2022). However, recent statistics indicate that the overall reduction in emissions has been negligible, with Gupta Strategists (2019) reporting a reduction of approximately 0% in 2017. Within the Dutch healthcare sector, building energy consumption accounts for 38% of the total CO₂ emission, according to Gupta Strategists (2019). Additionally, a study conducted by Heye et al. (2020) in a similar hospital in Switzerland revealed that imaging equipment, such as computed tomography (CT) and magnetic resonance imaging (MRI), contributed to 4% of the hospital's energy consumption. Furthermore, research by Shen et al. (2018) highlights medical equipment as the second highest energy consumer in hospitals in general, with CT scanners identified as one of the major energy-consuming devices.

Currently, MST does not integrate its CT scheduling system with data analysis, leading to a lack of insight into potential energy consumption changes, CO₂ emission reduction, and opportunities for creating a more sustainable hospital environment. Therefore, the primary objective of our research is to explore how connecting data with the CT scheduling system at MST can lead to improved energy efficiency, reduced CO₂ emissions, and a more sustainable hospital.

Approach

To address the objective, we tackled a scheduling problem concerning the two CT scanners operating at MST's radiology department. To effectively model this scheduling system, we utilized a Monte Carlo simulation implemented in Excel VBA. We classified the scheduling problem as an instance of the identical parallel machine scheduling problem, known for its nondeterministic polynomial time (NP) hardness (Pinedo, 2012). Our primary aim was to minimize the total makespan using the expected processing times. To achieve this optimization, we employed two key heuristics: the Longest-Expected-Processing-Time-first (LEPT) and Shortest-Expected-Processing-Time-first (SEPT) algorithms. From these results, we suggested a new scheduling system to reduce energy consumption and subsequently CO₂ emission. To create insights into energy consumption concerning scheduling, an energy measurement took place to calculate potential energy consumption reduction and a relation between the scheduling and the consumption.

Results

The simulation generated a week of scan requests with scan types and corresponding generated processing times, and we ran it for 25 replications for reliable results. This is a total of 125 days. We calculated the current average total completion time to be 40.6 hours. Using the LEPT and SEPT heuristics, the average outcome was 30.1 hours and 35.5 hours respectively with our calculated reserved scheduling blocks. Using LEPT, this is an improvement of 26%. Using the results of the energy measurement calculations, this translates into an annual reduction of 16473 kWh on CT1, which is 0.07% of MSTs annual energy consumption for this single machine and equal to the average energy consumption of three three-person households per year in The Netherlands (Nibud, 2023).

Conclusions and Scientific Implications

Our research findings demonstrate the potential of connecting data with the CT scheduling system at MST to improve energy efficiency, reduce CO₂ emissions, and create a more sustainable hospital environment. By utilizing the LEPT and SEPT heuristics, we have identified scheduling strategies that significantly improve average completion times, resulting in potential energy consumption reduction and CO₂ emission savings. LEPT outperforms SEPT by 14% and we therefore recommend a schedule to consist of the longest expected processing times first and the shortest expected processing times last.

Moreover, our study contributes to the broader scientific understanding of scheduling in stochastic environments, particularly within the context of healthcare facilities and the energy consumption of medical equipment. By exploring the relationship between CT scan scheduling and sustainable energy consumption, our research sheds light on the potential for optimized scheduling algorithms to drive environmental sustainability in hospitals.

Through our recommendations for measuring energy consumption, gathering relevant data, and implementing an efficient scheduling approach, we not only support MST in achieving its sustainability objectives but also offer valuable insights for the scientific community and the healthcare industry at large. The implications of our research extend beyond MST, providing a foundation for future studies on sustainable healthcare practices and the optimization of scheduling algorithms.

ACKNOWLEDGMENTS

Dear reader,

I have conducted this thesis to finalize my Bachelor of Industrial Engineering and Management. We have conducted our research at Medisch Spectrum Twente, a hospital in Enschede. The title is “Optimizing Computed Tomography Scan Scheduling for Sustainable Energy Usage: A Simulation Study at Medisch Spectrum Twente” and the research aims to assist Medisch Spectrum Twente in its sustainability goals and is a stepping stone for further research using a sustainability viewpoint within Medisch Spectrum Twente.

I would like to thank all my colleagues in the MedTech department for the great work atmosphere and for always being able to have a laugh or converse meaningfully, either on-topic or off-topic. I also am grateful for the guidance of my company supervisor Mark van Kooten for the time and effort to guide me through this process with all its complications and for always thinking along whatever the subject.

My gratitude goes out to my first and second supervisors from the University of Twente, Amin Asadi, and Erwin Hans for the guidance of the research and its content. Without your assistance, this venture would have taken quite a bit longer.

I hope that you will have a fulfilling experience as you delve into the information presented in this research.

Ward Schuppert, 25-05-2023, Enschede

CONTENTS

Management Summary	ii
Acknowledgments.....	iv
Contents.....	v
Glossary of terms	viii
1 Research Introduction.....	1
1.1 Company description	1
1.2 Research Motivation	1
1.3 Problem statement	2
1.4 Research goal	3
1.5 Research questions	4
1.6 Approach.....	5
1.6.1 Chapter 2: Problem analysis	5
1.6.2 Chapter 3: Literature review	5
1.6.3 Chapter 4: Solution formulation and selection.....	6
1.6.4 Chapter 5: Solution implementation	6
1.6.5 Chapters 5 & 6: Solution evaluation	6
1.6.6 Deliverables.....	6
1.7 Research Design.....	6
1.7.1 Research type.....	6
1.7.2 Data collection	7
1.7.3 Data analysis	7
1.7.4 Validity	8
1.7.5 Reliability.....	8
1.7.6 Limitations.....	8
1.8 Scope.....	8
2 Context analysis	10
2.1 Process description	10
2.2 CT Attributes	11
2.2.1 CT Operation	11
2.2.2 CT scan scheduling	12
2.2.3 CT scan types.....	12
2.2.4 CT regular versus urgent	13

2.2.5	Present scheduling method and schedule	13
2.3	Energy consumption	14
2.3.1	CT1 Energy Consumption	15
2.3.2	CTSEH Energy Consumption	17
2.3.3	Energy Consumption per Scan Type	17
2.4	Challenges and opportunities	19
3	Literature review	20
3.1	Previous literature on appointment scheduling accompanied by simulation	20
3.2	Modeling Approach	21
3.2.1	Scheduling	22
3.2.2	Our scheduling problem	23
3.3	Data analysis	24
3.3.1	Qualitative data	24
3.3.2	Quantitative data	24
4	Solution design	27
4.1	Quantitative data analysis	27
4.1.1	The observational study	27
4.1.2	Assumptions	27
4.1.3	Classification analysis	28
4.1.4	Cluster analysis	28
4.1.5	Cohort analysis	28
4.2	Scheduling Problem & Heuristic	28
4.2.1	CT Scheduling Problem	29
4.2.2	Heuristics	29
4.3	Monte Carlo Simulation	31
4.3.1	Data preparation	31
4.3.2	Data analysis	31
4.3.3	Simulation Input	32
4.3.4	Scan Generation	35
4.3.5	Reservation slots	36
4.3.6	Scan Scheduling	36
4.3.7	Replications	37
4.3.8	User Display	37

4.3.9	Schedule outcome	39
5	Study Results.....	41
5.1	Reserved scheduling blocks and makespan outcome	41
5.2	Energy Consumption Impact.....	43
6	Conclusions & Recommendations	44
6.1	Conclusions	44
6.2	Recommendations	45
6.2.1	Practical implementation.....	45
6.2.2	Future research	45
	Bibliography	48
	Appendices.....	51
	Appendix A: Interview structure.....	51
	Appendix A.1: Interview Questions	51
	Appendix B: CT Schedule & Codes	52
	Appendix B.1: CT Schedules	52
	Appendix B.2 : Scan Type Codes	55
	Appendix B.2.2 : CT Scan Type Codes Division.....	56
	Appendix C: Energy Measurement Setup	56
	Appendix D: Monte Carlo Simulation	57
	Appendix D.1: Processing Time VBA Code.....	57
	Appendix D.2: Scan Generation VBA Code	58
	Appendix D.3: Scan Scheduling + Calculation Procedures VBA Code	60
	Appendix D.4: Experiments VBA Code.....	72
	Appendix E: Simulation Input	74
	Appendix E.1: Scan types and their occurrence rate	74
	Appendix E.1: Replications.....	76
	Appendix E.2: Reserved slot calculations example.....	77
	Appendix F: Simulation Output.....	78
	Appendix F.1: Seven evaluated schedules and outcomes	78
	Appendix F.2: T-tests.....	79
	Appendix F.3: Urgent patient waiting time examples.	80
	Appendix F.4: Total makespan output examples.....	80

GLOSSARY OF TERMS

MST	Medisch Spectrum Twente
CT	Computed Tomography
MRI	Magnetic Resonance Imaging
In-house	Within the hospital
Care path	A care pathway is a complex intervention for the mutual decision-making and organization of care processes for a well-defined group of patients during a well-defined period (Vanhaecht et al., 2010)
VBA	Visual Basic for Applications
W	Watt
kWh	Kilowatt-hours
GWh	Gigawatt-hours
CO ₂	Carbon dioxide
HiX	Healthcare Information eXchange
LPT	Longest-Processing-Time-first
LEPT	Longest-Expected-Processing-Time-first
SPT	Shortest-Processing-Time-first
SEPT	Shortest-Expected-Processing-Time-first
LEPST	Longest-Expected-Processing-and-Setup-Time-first
MPSM	Managerial Problem-Solving Methodology
TBL	Triple bottom line
CT1	CT machine 1 used for normal patients
CTSEH	CT on the emergency department for urgent patients

1 RESEARCH INTRODUCTION

This section serves as a global introduction to the research performed at the hospital Medisch Spectrum Twente (MST) and is about sustainability with a focus on optimizing the energy consumption of computed tomography (CT) scanners. Section 1.1 provides more information about the hospital and Section 1.2 explains the motivation for this research. Next, Sections 1.3 and 1.4 define the problem and aim of the study. Whereafter, Section 1.5 introduces the research questions. Section 1.6 elaborates on the research approach taken. Section 1.7 explains the research design of this thesis. In the end, Section 1.8 provides information about the research scope.

1.1 COMPANY DESCRIPTION

MST is a top-clinical teaching hospital in the city of Enschede, which provides secondary and tertiary care. It originated from a merger in 1990 between two smaller hospitals in the city center. With around 3500 employees, including 235 medical specialists, and a catchment area of around 263,000 inhabitants it is also one of the largest hospitals in the area (Twente, 2022). The hospital has its main location in Enschede with a brand-new hospital newly opened in 2016. Other smaller outpatient clinics are located in Oldenzaal and Haaksbergen.

MST is an education center for future doctors and conducts a lot of scientific research. They have close ties with the University of Twente regarding education, training, and research, and develop technological innovations together. Recently, MST has signed a green deal and the hospital tries to improve itself in its sustainability field. However, the deal only talks about goals and not about how MST achieves these goals or what consequences there are when not achieving them.

This research falls under this broad goal and aims to provide MST insight into this sustainability area. It also provides an example of how to break this broad goal down into solvable topics.

1.2 RESEARCH MOTIVATION

The healthcare sector is a sector where constant pressure lies on the level of care. Healthcare institutions must maintain the key principles of innovation and development to continuously improve the service level provided. In their yearly report, MST has set seven critical success factors to continuously improve their overall performance. One of them is *“sustainability, care, education and science”* (Medisch Spectrum Twente, 2022). They do this by embracing new technologies, re-evaluating current technologies, and introducing long-term plans to reach their goals.

MST has set sustainability goals to reach before 2030 and 2050 in line with the Dutch government's requirements. The government set targets such that CO₂ emission must be 49% lower in 2030 than in 1990 (Rijksoverheid, 2022). Mahase (2022) has found during her research in the United Kingdom, Ireland, and the United States that in general, the workload is already quite intense within hospitals and employees may not be looking for more time-consuming work to find out how to incorporate sustainability in their daily activities. Therefore, MST has appointed a sustainability coordinator recently and is working on a hospital-wide policy plan regarding sustainability and these goals, and how to tackle various problems. Without such a sustainability manual, the set goals will be challenging to attain, and potential consequences can be disadvantageous.

The triple bottom line embodies the desire of MST to reach its sustainability goals and keep its top-clinical status. *“The triple bottom line (TBL) of sustainability is an important emerging conceptual framework which considers the combined economic, environmental, and social impacts of an activity”* (Vergunst, et al., 2020, p. 48). Sustainability is becoming a greater topic within organizations and the triple bottom line will become increasingly significant as well. Economic value and social impact are great indicators to assist the realization of sustainability goals, hence we see the TBL as a business tactic to support their set targets. Therefore, MST can significantly improve its current position in the sustainability field by implementing economic value and social impact in its sustainability approach.

1.3 PROBLEM STATEMENT

Research shows that the energy consumption of buildings causes nearly 40% of the total CO₂ footprint in the healthcare sector in the Netherlands (Gupta Strategists, 2019). At a certain comparable hospital in Switzerland, where there are three CT and four magnetic resonance imaging (MRI) scanners, the energy consumption of those appliances accounted for 4% of the hospital’s total energy use (Heye et al., 2020). Bawaneh et al. (2019), Friendly Power (2021), and Shen et al. (2018) show that the largest energy consumers in certain hospitals and in general are the heating, ventilation, and air conditioning (HVAC) systems and lighting systems. Additionally, Shen et al. (2018) mention medical equipment as one of the larger consumers. MST is already performing research on the HVAC systems and lighting systems. The next sensible energy consumption problem to tackle lies in the medical equipment area.

Friendly Power (2021) suggests that imaging equipment is *“a good candidate for improved efficiency because it draws a lot of power when not in use and is in use only in short bursts”*. For MST, there has been no prior research concerning the energy usage of their imaging equipment. Heye et al. (2020) show that CT and MRI scanners account for 4% of the total energy use in a similar hospital and Shen et al. (2018) show 8% of the total energy consumption in a particular hospital is from medical equipment alone. From this, we conclude that CT & MRI scanners should be responsible for a relatively high percentage of energy consumption per scanner in comparison to other energy consumption per system. Where MRI scanners always must be in the on-state and cooled to maintain the magnetic field necessary for scanning, CT scanners can be turned off and have more room for improvement in this field. This could mean that within the CT equipment, with relatively low effort, we could realize a relatively large amount of improvement. A low percentage decrease can therefore be a large absolute decrease in total energy consumption. We have found that research in this area is impactful; there is a huge opportunity for improvement that can lead to significant energy reduction. Moreover, there is a gap regarding the knowledge of the optimization possibilities within MST concerning their appliances, which also is encouraging to research this area.

The whole healthcare sector still has enough work to do concerning sustainability. The CO₂ emissions remain at the same level in comparison with 1990 (Gupta Strategists, 2019), and time is running out. The city of Enschede is putting great effort into becoming a more sustainable and greener city, and the MST cannot stay behind. Hospitals are significant contributors to environmental change, specifically regarding their direct energy consumption (McGain & Naylor, 2014). Within MST, several employees including a biomedical technologist, a radiologist, and a maintenance engineer have noted that there is no current knowledge of healthcare-specific energy usage. This is due to them measuring electricity on a larger scale, namely story wide instead of per floor or operating room. When the new establishment Koningsplein was built, MST did not see the use of extra gauges in the electricity network and since the investment was not deemed valuable, they did not build them into the energy grid. This also means separate floors are not

that easily measured. Only machines where we can place a simple gauge in between would be easily measured.

MST is working towards its goal of CO₂ reduction, but they do not currently connect potential changes toward more sustainable healthcare with hard statistics that could be readily available within its systems. In the past, MST has created a detailed schedule per working week for its CT in the radiology department (CT1) and its CT in the emergency care department (CTSEH) including a ghost planning to slip non-urgent patients in between emergency patients (CTBLANCO). The choices made for the implementation of the schedule have been based on throughput amongst others, but not on sustainability or energy usage. That brings us to the research goal.

1.4 RESEARCH GOAL

There is little knowledge of what influence improvements within the department have on their sustainability goals, negatively or positively. Figure 1 shows the cause-and-effect relationships in the problem cluster.

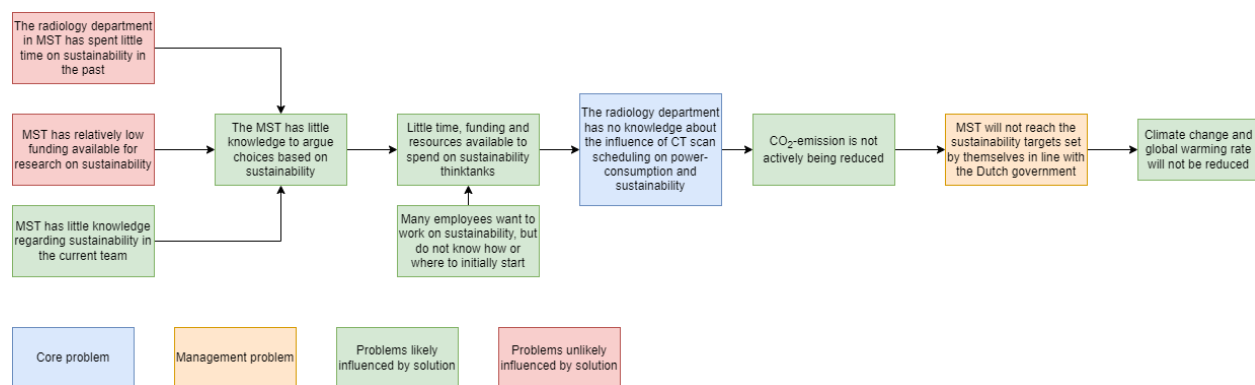


Figure 1: Problem cluster (own figure, information gathered from interviews)

The focus of this research lies on the two CT scanners in the radiology department, where we investigated the scan scheduling influence on energy efficiency. We defined energy efficiency in this context as increasing the degree of utilization per time period, resulting in less energy consumed during nonproductive idle states and more energy spent on actual scanning (Heye et al., 2020). To perform our research, we made use of a common problem-solving method.

Heerkens & Van Winden (2017) mention that a core problem is a gap between the norm and reality. We expressed this problem in singular or several variables to make it quantifiable. The reality now is that MST does not have a clear view of how much energy their scanning equipment, explicitly the CT scanners, consumes and what influence different choices have on the energy consumption of these machines. With the knowledge about how much the equipment consumes in various states or during various scans, a simulation study can provide insight to reduce the energy consumption of the machines. Incorporating the key performance indicators, we mapped and tweaked the current situation to improve efficiency and sustainability. The norm for MST and this research is to provide insight into energy consumption and possibilities to lower it while not lowering the quality of service for patients. The reality is that there is no insight into energy consumption behavior concerning scan scheduling. To narrow the gap between the norm and reality in reducing energy consumption, we proposed the use of Monte Carlo (MC) simulation together with single or multiple heuristics within MST and the radiology department. Specifically, the MC

simulation is capable of simulating the current scheduling system which includes stochastic data such as scan duration for instance. MC simulation is particularly useful when dealing with uncertainty and stochasticity (Shapiro et al., 2000; Shapiro, 2001; Birge & Louveaux, 2011). A heuristic is then necessary to optimize the scheduler, taking an alternative approach to the current system. By comparing the results of different scenarios, we identified significant differences. It is worth noting that the inclusion of heuristics is vital to improving the current system and facilitating optimization. Given its ability to support scenario analysis and provide optimal outcomes, we have chosen the MC simulation as our preferred method of simulation.

Because there is no prior knowledge about this subject within MST and the energy consumption per CT scanner is relatively high, there is a significant opportunity to optimize the energy consumption of these types of scanners. This research aims to lower energy use and ultimately lowering CO₂ emissions and become more sustainable. To do so, we sought ways to optimize scan scheduling and machine use. We split the research into multiple research questions to delineate the goal into manageable parts.

1.5 RESEARCH QUESTIONS

We derived a main research question based on the core problem of MST. MST does not immediately know the impact of changes in scanning equipment in the radiology department on sustainability and energy savings. For this research, we focused on changes concerning the scheduling of operations of CT scanners. Simulations with either real-time data or validated data help in gaining knowledge about this question. The research goal was to help MST be more sustainable by giving insight into the energy consumption of CT scanners. The research also attempted to reduce the energy consumption via optimized scan scheduling. An optimized schedule leads to less system idle time during the system on-state, which is the state where the machine is turned on, but not performing work, resulting in less energy lost. To assist this goal, we have used a Monte Carlo simulation and constructed research questions that we have answered during the research. This section contains the main research question and sub-questions.

Research question: *How can the department of radiology of Medisch Spectrum Twente, optimize its machine scheduling for sustainable energy- and cost-saving opportunities for CT scanners?*

We subdivided the main research question into several sub-questions to create practical research. The questions create a better structure for the research and provide a better view for the reader of the report. We divided the main question into these:

1. What does the current scheduling of the CT scanning equipment look like?
 - 1.1. How can this data be derived?
 - 1.1.1. What assumptions have to be made for non-existing data?
 - 1.2. Who are the stakeholders?
 - 1.3. What is the current performance?
 - 1.3.1. What are the key performance indicators?
 - 1.3.2. What are the zero measurements of these indicators?
 - 1.4. Is the scheduling influenced by people or only by regulations?
2. How can the current scheduling of CT scanning equipment be optimized to increase energy efficiency while keeping the same service level for patients?
 - 2.1. How can the scheduling problem be formulated?
 - 2.2. What heuristic should be used?

3. What is the energy consumption of the CT equipment with the current scheduling process?
 - 3.1. What is the idle time and usage?
4. What recommendations can be made for sustainable cost-saving opportunities within the radiology department?
 - 4.1. How can the department improve machine use scheduling for a more sustainable future?
 - 4.2. What are the main findings from the simulation?

1.6 APPROACH

Keeping the main research question in mind, the first steps for solving the problem are quite clear. It is required to define the problem and acquire missing knowledge and information. We have used the Managerial Problem-Solving Methodology (MPSM) by Heerkens & Van Winden (2017) throughout this research as a directive as it is a comprehensive guide for graduate researchers to arrive at solutions using a systematic approach. The MPSM consists of seven steps:

1. Problem definition
2. Approach
3. Problem analysis
4. (Alternative) solutions formulation
5. Solution selection
6. Solution implementation
7. Solution evaluation

Section 1.3 covers step 1. This subsection covers step 2. We further expanded on steps 3 to 7 in the following subsections, namely 1.6.1 to 1.6.5, also corresponding with the chapters where we have addressed them.

1.6.1 Chapter 2: Problem analysis

We have looked into the process and technology around the CT scanners to properly analyze the problem and current situation. We analyzed how the scheduling process is constructed, monitored, and regulated. We also have analyzed KPIs and corresponding data to create an overview of what is already accomplished regarding CT within MST and to determine how performance currently is measured. Then, we made observations and held interviews to combine all the gathered information and to process the information in good order.

1.6.2 Chapter 3: Literature review

Next, to come to conclusions and gather the information and knowledge necessary for this research, we performed a literature review. We have solved and answered knowledge problems such as research question 2 by looking into previous research done on the subject. Making assumptions, finding KPIs, and evaluating measured or derived information from other research questions can all be argued using information gathered by literature review. To perform the literature review, we consulted the databases Google Scholar, Scopus, and Web of Science.

1.6.3 Chapter 4: Solution formulation and selection

Then, using a literature review, we found heuristics to solve the scheduling problem of the CT equipment, which we modeled as a mathematical model. We have selected and reasoned tools to assist the application of the heuristics. We applied various KPIs and tested and evaluated different scenarios.

1.6.4 Chapter 5: Solution implementation

We have implemented the chosen heuristics from the literature study. To do so, we made assumptions and underlying simplifications. This is to create an effective and structured simulation. We have verified the assumptions and simplifications with stakeholders from MST. Later, we determined the input for the chosen heuristics. The variables and data we have used as input involve the information found in sub-question 1.3. We have used the output of the model via the heuristic to evaluate the proposed heuristics. Next to that, we validated and verified the heuristics to ensure the outcome is relevant.

1.6.5 Chapters 5 & 6: Solution evaluation

Ultimately, when the final heuristics and assisting mathematical model are ready, we evaluated the capacity of the model and its outcomes to describe real-world expectations and knowledge. Then we compared the output of the simulation with the status-quo results. We aimed that the insights derived from our simulation could increase the energy efficiency of scanners which is useful for MST. If MST finds the outcome of the simulation a practical one, they can implement it. If that happens, we could compare the results of the implementation with the results from the simulation, to see if the model describes reality. Based on the results of the research, we concluded and gave recommendations regarding future research and improvement.

1.6.6 Deliverables

The research we conducted at MST aims to create a more sustainable work environment with more insights into energy consumption and an improvement in the energy efficiency of the CT scanning equipment. To accomplish this, we researched to improve the scheduling process of the CT and the energy consumption. With the use of the intended heuristic, per the norms, values, and wishes of MST, we recommended changes in the CT scheduling process. We delivered this together with the simulation itself for potential future use or for it to be applied elsewhere with modifications.

1.7 RESEARCH DESIGN

During this study, we used several research methods to get a clear overview of the whole subject and answer the research questions. This section of the report explains what type of research we conducted, what the research population is, the gathering and handling of data, the validity and reliability, and lastly, the limitations.

1.7.1 Research type

This thesis' study includes three categories: descriptive, explanatory, and evaluative. The first category contains the analysis of the problem to gain a better understanding of the current state of affairs at the time of the research. This also involved gaining better knowledge about the manner of working in MST and the current situation in the radiology department regarding energy consumption and sustainability. To obtain this information, interviews, observations, and a literature review took place. We conducted interviews with lab technicians, radiologists, and other administrative personnel working with the CT. Observations on the workplace at the CT identified the design structure of the scheduling.

In the explanatory research, we examined the data and formulate, chose, and implemented the solution. Explanatory research determines “*how variables interact with each other and [identifies] cause and effect relationships*” (Study.com, 2022). In this part, we figured out how the department of radiology can be more sustainable regarding our subject.

Since the research aims to find out several possible choices to make about sustainability, based on the needs of MST, evaluation is necessary to ensure that we achieve the research goal. We accomplished this by performing evaluative research.

1.7.2 Data collection

For the research, we have used the available data from MST and their departments. The data collection encompasses qualitative and quantitative data. Examples of qualitative data are why and when certain scans are performed and scheduled. An example of quantitative data is the energy consumption of a certain scan in this case. Since this is not the first research on this general topic, we gathered much information through previous literature to provide knowledge to assist the research as mentioned in 1.6.5. Next, we explain the two data categories.

We collected qualitative data, such as background knowledge about the quantitative, relatively straightforwardly. We performed interviews with CT users with the purpose to gain knowledge about the CT equipment technology-wise and how, why, and when the CT is used. For this research, this type of data mostly was background information to understand the quantitative data.

Concerning the quantitative data, we performed several real-life measurements of the equipment’s energy consumption during a certain time span. The company behind the equipment provided technical details about the equipment useful for the simulation of energy consumption. Data regarding patients’ scheduling was available in MST’s information system HiX. In addition, we monitored the arrival, scanning, and other relevant processes for two days to gather relevant data and know whether it is reliable data. Thereafter, we consulted HiX to combine gathered data with other available, validated data. All this data is available for use in the simulation.

1.7.3 Data analysis

Between the different types of data is also a different type of analysis. We assessed qualitative data by considering all gathered opinions and attitudes towards certain ideas and coming to a general conclusion. General patterns in equipment usage that tend to be wasteful became known and we processed them into the recommendations of the research. This pointed to some starting points for the qualitative data analysis.

We required much quantitative data in the hospital system, which have different data types. For example, scan types, scan occurrence, and equipment idle time. We also had to know how long certain scans take in general and why scans can differ in duration. To analyze and combine all the data, it was clustered. Continuously looking through raw, scattered data is more time-consuming than having clustered data in well-ordered places. We needed to combine multiple data files with distinct characteristics containing information about certain scans for a clearer, overarching overview. When we analyzed and clustered the data, the next phase started where the simulation model takes shape.

1.7.4 Validity

Research needs to be evaluated to be deemed valid. We needed to evaluate the created model and its outcome to determine whether the quality of the model is high enough and the outcome sufficient. An article from Heale & Twycross in 2015 covers validity and reliability in quantitative studies. '*Validity is defined as the extent to which a concept is accurately measured in a quantitative study*' (Heale & Twycross, 2015, p. 66). This means that the scope of a research should be clearly set based on the goal and requirements of the study. For example, in this research when looking at the energy consumption of the scanning machinery, the research should not incorporate the energy consumption of the creation of the machinery, since that is not of interest to MST and changes the course of the research toward its goal.

1.7.5 Reliability

Reliability is '*the extent to which a research instrument consistently has the same results if it is used in the same situation on repeated occasions*' (Heale & Twycross, 2015, p. 66). In other words, if other scholars conducted the same research, we would expect the outcomes to correspond with each other. It is important to note that reliability does not necessarily guarantee validity; it simply means that the research instrument is consistent in producing similar results. As long as the choices made during the research are grounded in sound reasoning, we can consider the research to be reliable.

1.7.6 Limitations

One of the main limitations of the research was the lack of real-time data. As mentioned previously, during the construction of the new building, MST did not see the use of adding measuring gauges between smaller levels within the hospital. This means that we could not measure the scanning equipment for a whole week or longer for example to compare energy data with historical data and see scheduling change influences. Also, near to no data existed within the MST on this subject. We used data from similar research for input as a result of which our research becomes more theoretical.

MST's information system HiX was introduced at the end of 2021 and therefore data logs were different compared to past data. Gathering and comparing data is more difficult, since logged parameters in the past, such as timestamps, are not clear if they log the entry of the patient, scan start, or scan end for example. More time is, therefore, necessary to combine all data.

Furthermore, the timeframe available for this research was a limitation. When making decisions, we made concessions. This is because, within the time span, we could only gain a certain amount of knowledge. Due to the time limitation, we did not incorporate the implementation of the recommendations into this research to lay focus on the soundness of the research.

Lastly, a limitation of this research was that we can only examine and modify a certain angle of view of the user side. This is due to either lack of knowledge of machine specifics and scan parameters for their respective goals or due to the user not being able to influence certain energy-consuming factors (i.e., a better idle or standby state).

1.8 SCOPE

This study focuses on the energy usage and scheduling of CT1 and CTSEH scanners within the hospital MST in Enschede. We measured energy use and made that information available to support decision-making in MST. The scheduling included the planned and emergency use of the two CTs of the radiology department and the associated employees in the process. We conducted the research itself in

approximately three months and the research aims to create insight into the energy usage of the CT and to increase the energy efficiency of CT use. MST can review the implementation of the recommendations to investigate the viability of the scheduling optimization. Next to that, we evaluated the influence and importance of the gained insights. We did not incorporate the MRI scanners in this research due to distinctive characteristics that influence scan duration and scheduling and time limitations.

2 CONTEXT ANALYSIS

This section contains the evaluation of the current situation within MST regarding its CT scanners. Section 2.1 explains the CT process in general terms. Section 2.2 elaborates upon CT scheduling and scan types. Section 2.3 dives deeper into the energy consumption of the CT scanners. We have gathered all data through interviews and observations.

2.1 PROCESS DESCRIPTION

Computed tomography is a medical imaging technique where detailed images of the body are produced by putting a patient in a rotating X-ray tube. The tube then rotates through the scan gantry to measure the X-ray attenuations caused by differences in the tissue within the body. These multangular X-ray images are processed to create cross-sectional images of the body. MST has two scanners in the radiology department, CT1, and CTSEH, where SEH is an abbreviation for the Dutch ‘spoedeisende hulp’, meaning emergency department. Both scanners are Siemens SOMATOM Definition Flash. In Table 1 we see the scans per specialism on CT1 and CTSEH.

Table 1: Number of CT scans per specialty on CT1 and CTSEH in 2019 (derived from data MST, 2022)

Specialism	Requests CT1	Requests CTSEH
Anesthesia	0	3
Cardiology	836	334
Cardiopulmonary Surgery	76	116
Clinical Neurophysiology	0	3
Dermatology	0	1
External specialism	130	29
First Aid	0	2
Gastrointestinal and liver diseases	832	634
General Practitioner	99	561
Gynecology	132	135
Intensive Care Specialist	51	523
Internal Medicine	1727	1449
Lung diseases	1006	2525
Neurology	147	4539
Neurosurgery	36	416
Ophthalmology	1	13
Oral and maxillofacial surgery	15	28
Orthopedics	41	760
Otorhinolaryngology	355	922
Pediatrics	7	48
Plastic Surgery	29	24
Psychiatry	1	10
Radiology	3	3
Radiotherapy	18	12
Rheumatology	6	21
Surgery	1022	3757
Unknown	0	1
Urology	519	714
Total	7089	17583

Certain specialisms request the largest amount of CT scans: neurology, surgery, lung diseases, and internal medicine. Figure 2 shows these four groups respectively to the other specialisms.

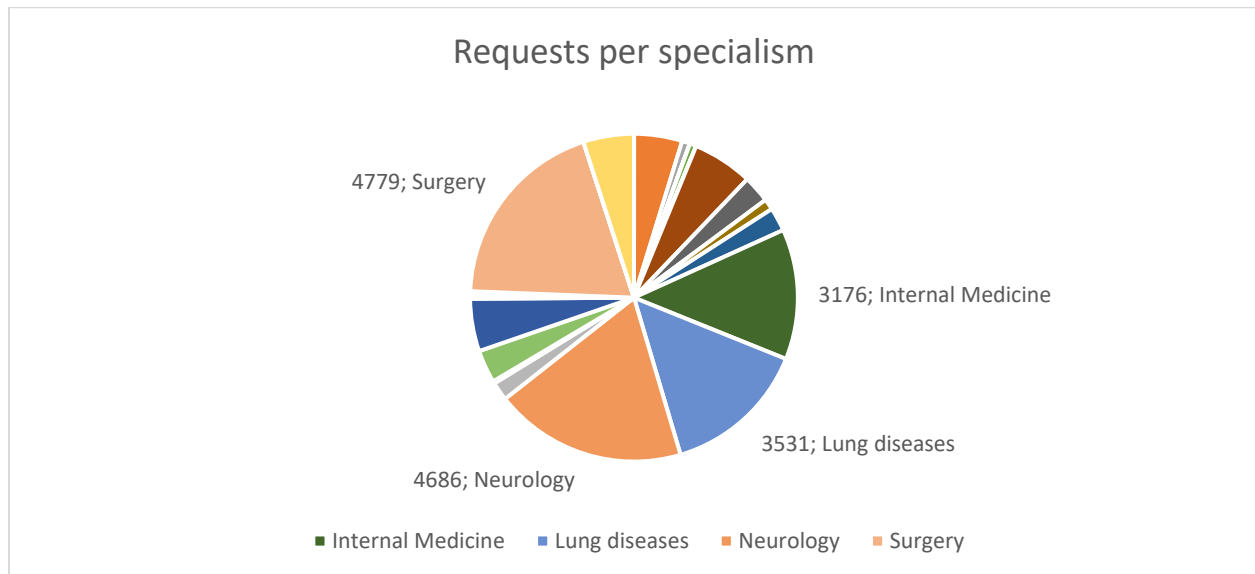


Figure 2: Top four specialisms regarding CT scan requests in 2019. Total = 24672 (derived from data MST, 2022)

Regular patients arrive at the respective administration to register themselves whereafter they are placed into the waiting room for the CT1 or CTSEH. Thereafter the patient is called to the CT, prepares themselves, and is placed in the CT. The intravenous line for contrast fluid is administered to the patient if necessary and then the (scout) scan is performed. After the completion of the CT scan, the patient generally remains in the scanner until the lab technicians have reviewed the scan. If the scanned image is not deemed clear enough for a review, the scan is retaken. If all went well, the scan is sent through the information system to the radiologist for review.

2.2 CT ATTRIBUTES

In this section, we discuss the CT attributes that are part of this research. Subsection 2.2.1 discusses the CT operating hours and policy, and we continue in Subsection 2.2.2 with the CT schedule. Subsection 2.2.3 elaborates upon the different CT scan types and their attributes, whereas Subsection 2.2.4 continues with the difference between regular and urgent patients. Subsection 2.2.5 expands on the present scheduling method and how the schedule is filled.

2.2.1 CT Operation

The CTs of the radiology department are operated differently from each other. Operating hours for CT1 are from 08:00 up to 16:30, but CTSEH works continuously. The shifts on the CTSEH are from 08:00 to 16:30, 16:30 to 23:00, and 23:00 to 08:00, where the day shift has two lab technicians and the evening- and night shifts only have one lab technician.

Deceuninck et al. (2018) coined an appropriate policy name that describes the system in MST. During observations, we found that the lab technician always chooses the patient in the waiting room with the lowest appointment number. In reality, if the time is 09:40 and patient X has an appointment at 10:00 but has arrived early at 09:30 and patient Y has an appointment at 09:50 and arrives early at 09:45, patient Y

is chosen as the next patient to be treated, so the patient with the lowest appointment number is chosen. This policy is called the Appointment Order Work Conserving (AOWC). MST has one alteration to the AOWC. If patient X already is waiting and patient Y has not yet arrived, they will treat patient X instead of waiting until the appointment time of patient Y has passed.

2.2.2 CT scan scheduling

For the appointment scheduling process, we distinguished three types of patients: standard, emergency, and walk-in. Standard and emergency patients can either be in- or outpatient, but walk-in patients are always outpatients.

When a patient requires a CT scan, a doctor requests one through the internal software system in MST. A radiologist then reviews the request before arriving in the scheduling system. Once approved, the appointment arrives in the scheduling system. The patient is then allocated to an appointment slot in adherence to their respective scan type and expected duration. At the CTSEH, patients are only scheduled until 10:00 in the morning for 5 minutes so-called *blank* scans to increase utilization of CTSEH, since the past had shown that the CTSEH is not constantly filled with emergency patients during the day. Blanco scans are scans that do not use contrast fluid. Walk-in patients typically also have scan types with low processing times such as *blank* scans. Figure 3 shows us the timeline of the CT scheduling process.

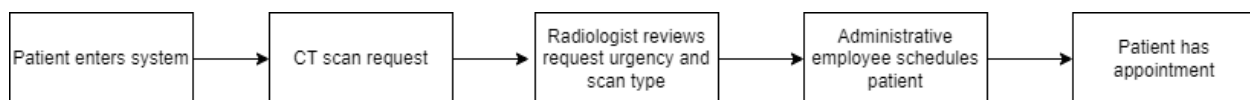


Figure 3: Patient CT schedule flow in MST for all care paths on the CT1 and CTSEH (own figure)

2.2.3 CT scan types

There are several different scan types performed on the CT equipment. In 2019, 119 different scan types have been performed with a total of 24673 performed scans on an equal number of patients divided over CT1 and CTSEH. The scan types range from small scans that only scan an ankle to large scans such as full body scans. Another influential factor in the duration of a scan is if contrast fluid needs to be administered or not for imaging reasons. A patient can either have small blood vessels, which makes it difficult to find the right place to pierce the skin with the needle. Hence, the scan time, therefore, can vary heavily, especially for the scans with lower duration.

The average scan duration of different scan types ranges from 5 to 30 minutes, whereas the most time-consuming scans even take up to an average of 60 minutes. Scan types or SPS codes, denoted with C-codes, and their respective average duration can be found in Appendix B.2.1. Scan duration is not influenced by scanning on CT1 or CTSEH since the machines are identical.

The arrival rate, or request rate, of CT scans, was not available. There are multiple streams in the scheduling system for a scan to arrive where doctors and administration employees can both assign patients to an appointment slot. The requests per time period were therefore difficult to manually monitor and are not logged in the scheduling system. Subsection 2.2.5 elaborates more on this matter. Figure 4 depicts the occurrence rate of all scan types in 2019. There are only 20 scans with an occurrence rate of >1%.

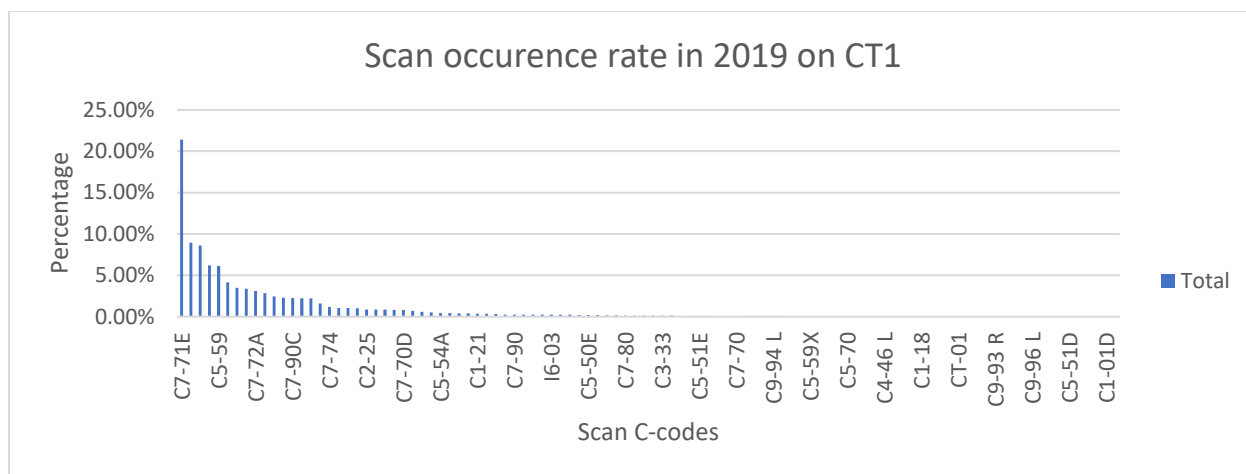


Figure 4: Scan type occurrence rate in 2019, Total = 7089 (MST, 2022)

Scan processing time data is also not available. Data from before HiX showed the following:

- scheduled appointment time.
- scheduled scan duration, and scan start time.
- when the patient has been registered in the waiting room.
- when the patient has left the CT room.
- a moment that could not be verified what the specific time represented.

2.2.4 CT regular versus urgent

Despite the literal meaning of the term *emergency* that suggests only first aid patients would be classified as urgent, other patient classes may also be classified as urgent patients. For instance:

- in-house patients, meaning patients who are already admitted to the hospital, from various clinics.
- patients from consultation appointments that have been forwarded to the CT.
- patients from care paths that are set the day itself.

All these classes are deemed urgent, since all requested scans that are not planned for at least a day later, are urgent.

2.2.5 Present scheduling method and schedule

Patients enter the scheduling system either as in- or outpatients. Within these two ways, there are two more states: centralized and decentralized. Centralized means that appointment requests enter the system through the administrative office of the radiology department. Decentralized means that an appointment is scheduled from a different department than the radiology department. A difference between these two influxes of appointments is that decentralized influx is planned and then enquired for scan protocol preparation by a radiologist, while centralized is first enquired from scan protocol preparation before being put into an appointment slot. This can require less redundant effort, since a radiologist can assess a scan request differently, causing the scan protocol or scan preparation to increase in time, requiring a larger appointment slot and requiring the rescheduling of the. On the other hand, other departments should be able to schedule a patient together with the patient right away. To give a

radiologist enough time to protocol a scan, decentralized planning can only happen up to 5 working days in advance.

Once a patient requires a CT scan, the doctor or practitioner in question enters the patient into the scheduling system. The practitioner then describes the patient's status and what they want to assess. Next, the request arrives at the radiology department. Here, a radiologist evaluates the request and adjusts the scan type if required. Once they give urgency to the request, they pass it on to the administration of the radiology department. This department has the task of scheduling the requests. They have insight into the schedule and assign the appointment request in the designated schedule slot.

Within MST, these allotted slots are predetermined. Certain scan-type slots are only set at specific moments in a day and not every single day. Several appointment slots are kept available up to three working days beforehand for urgent patients. Other appointment slot restrictions are generally lifted three working days beforehand to attempt to fill the schedule regardless of other scan types. There are three different schedules: CT1, CTSEH, and CTBLANCO. CT1 is filled with timeslots and has some room for delay. CTSEH is not planned in advance, but potential walk-ins or *urgent* patient types will be put in the system here. CTBLANCO has appointment slots till 10:00 for *blank* scan types, which are 5-minute scans, to be scanned on the CTSEH.

Patients requiring a CT scan at the radiology department are divided over the three different schedules accordingly. The CT1 schedule consists of all scan types and is divided into different scheduling blocks with different grouped scan types. The CTSEH schedule has alternating 15-minute blocks till 10:00 for either scan that takes less than 15 minutes or empty blocks reserved for urgent scans. The CTBLANCO schedule has the same principle and is a dummy schedule for the CTSEH. It has alternating scheduling blocks to either keep the schedule free for emergencies or 5-minute blank scans. The latter is to place for example walk-ins in the schedule to keep the CT occupied. Appendix B.1 shows all current appointment slots for the different schedules.

2.3 ENERGY CONSUMPTION

During the construction of the new Koningsplein building of MST in Enschede, MST chose to only be able to measure energy consumption per energy branch. This unfortunately means that there is no secluded data readily available for only the CT equipment. An energy branch can be seen as a vertical energy pipe running from the roof to the basement and where groups are connected to. Figure 5 shows an example. Here the blocks are floors of a building, and the small squares are equipment in need of energy. The lines are powerlines that form a figurative branch together and then are the example of an energy branch that can only be measured as a whole with the current grid management.

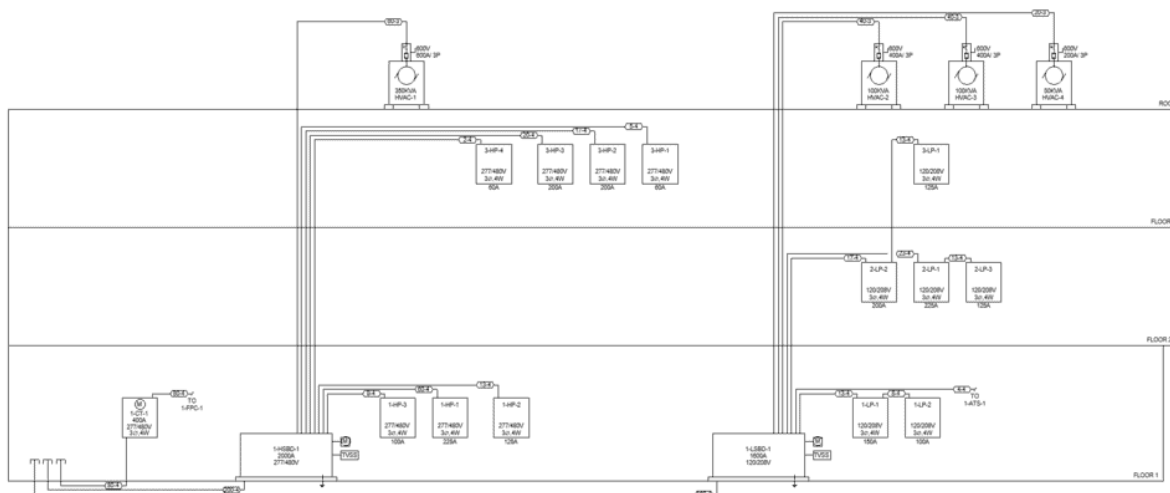


Figure 5: Example of energy branches (Types of Electrical Drawing and Diagrams, 2022)

In cooperation with Kuno Kamstra, a maintenance engineer within MST, we have measured the energy consumption during a certain week on the CT1. Appendix C shows the applied setup. The measuring device to measure energy consumption is a Janitza UMG 511 Power Analyzer. We have measured the data in kWh with values per minute due to the equipment erroring when trying to retrieve data in kWh values per second. By linking the energy consumption with the performed scans and making assumptions, we generalized the energy consumption per scan type. This section elaborates on the CT1 and CTSEH energy consumption and the energy consumption per scan type.

2.3.1 CT1 Energy Consumption

Figure 6 shows the CT1 energy consumption in kWh. We performed the measurements over a total of approximately 7 days plus one hour from the 18th of October 2022 at 09:49 up to the 25th of October 2022 at 10:42. The graph shows the accurate energy consumption on the y-axis for every minute on the x-axis.

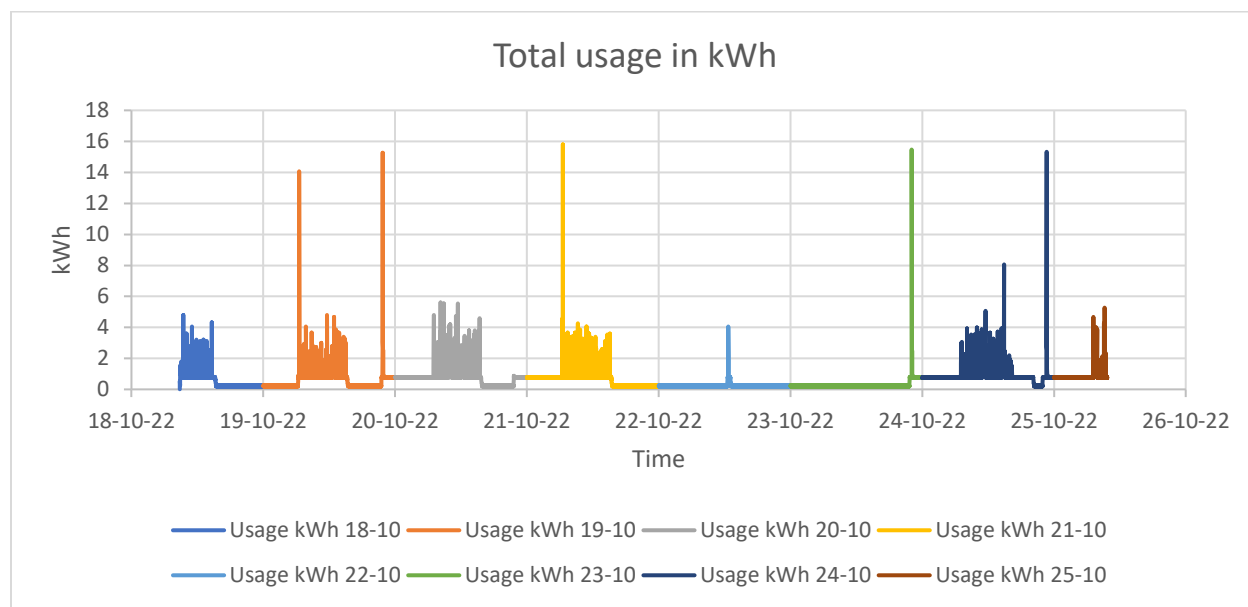


Figure 6: Total usage in kWh (MST, 2022)

We found that the average values for the system off-state are 12 & 16 kWh and for the idle state 44 & 48 kWh. The CT requires a large amount of energy during startup, which is on average 15.2 kWh the first minute and the next six minutes an average of 2.67 kWh per minute before returning to the average idle state of around 0.8 kWh per minute. Over this time period of a week plus one hour, CT1 has consumed an amount of 6270.5 kWh including the weekend without elective scans, meaning scans that are chosen rather than urgently necessary. The idle state has consumed 3046.2 kWh total during this time period. That is 49% of the total energy consumption.

Figure 6 shows, in this particular week, that the machine has also been idle instead of off for four nights, which has been a human error and a large addition to the idle state usage. For comparison, CT startup consumes around 37 kWh, and leaving the CT idle during the night instead of off consumes respectively 690 kWh instead of 210 kWh. If we extract the idle time during the night, the idle state during operation hours has consumed 1053.6 kWh. This is 16.8% of the total energy consumption and 38.8% of the total energy consumption during operating hours. Table 2 shows these numbers.

To calculate the total energy consumption of CT1 during operation, we multiply the measured energy consumption from these 5 working days times 52 working weeks. Keep in mind that the energy consumption data during operating hours do not include the nighttime idle state consumption. Unfortunately, we could not verify the scan types performed during this week with the occurrence rate per scan type due to these scan types only being able to be retrieved in a different format than SPS codes and linking them both being non-viable.

Table 2: Measured and calculated energy consumption

Measured and calculated energy consumption		
Total energy consumption for 7 days (measured)	6270.5	kWh
Total idle consumption for 7 days (measured)	3046.2	kWh
Total energy consumption during operating hours per 5 days (measured)	2715.0	kWh
Total idle consumption during operating hours per 5 days (measured)	1053.6	kWh
Total energy consumption during operating hours per year (extrapolated)	141180	kWh / year
Total idle consumption during operating hours per year (extrapolated)	54787	kWh / year
Idle state per year during operating hours	38,8%	

The total electricity usage of MST is 22.6 GW per year (MST, 2022). We have retrieved an estimate of the total CT1 energy consumption by adding the average off-state consumption outside operating hours including the weekend to the total energy consumption during operating hours. We deemed this a low estimate since CT1 can already be seen left on for four nights during a measurement of seven nights. Non-operating hours from 16:30 to 07:30 total 15 hours per night and Saturday and Sunday are 24 hours each. This should result in 123 hours of off-state energy consumption every week. Off-state consumption averages 14.4 kWh, thus we assume the average total energy consumption per week to be 1771.2 kWh. This results in a calculated average of 92102.4 kWh per year. The average energy consumption of CT1 totals a calculated sum of 233282.4 kWh per year, which is 1.03% of MST's total energy consumption per year for only one single machine.

2.3.2 CTSEH Energy Consumption

Because we could not realize another measurement of CTSEH within the scope of the assignment, we took a theoretical approach to calculate its energy consumption. CTSEH is never shut off completely, due to its necessity of being readily available for an emergency scan. Due to this, it will never have the off-state consumption of 12 or 16 kWh, but it always has 44 or 48 kWh of idle-state energy consumption while idle.

In 2019, CTSEH had 137695 minutes scheduled for performed scans (MST, 2022). This includes emergency scans, which are also put in the system with a scheduled scan time. That means that in general there are at least 387905 minutes of the idle state outside of the scan processing time. The average idle consumption between 44 and 48 kWh translates into 0.767 kWh per minute, which means there is at least 0.29737 GWh of idle state energy consumption on CTSEH in 2019 in general.

To calculate the energy consumption of the scanning itself, we have used the scheduled duration of the scans and an average of the energy consumption per hour of the measured scanning state of CT1. We notice a relatively larger number of abdominal and thorax-type scans on CT1 and a relatively larger number of cerebrum or cervical vertebra (CWK) scans on CTSEH during data analysis. Abdominal and thorax-type scans require more energy than cerebrum or CWK scans, but the latter scan types also take less time in general (Mohammad, et al., 2011). The difference in energy consumption between these scan types is therefore not of considerable influence on these calculations.

To calculate, 137695 minutes of scan time translate into 2294.92 hours. The average measured energy consumption per hour of full-time scanning on CT1 is 66.47 kWh. The scheduled scan period of CTSEH consumes $2294.92 * 66.47 = 152543.3 \text{ kWh}$ on average using the scheduled scan time. The total calculated energy consumption of CTSEH in 2019 is then $152543.3 + 297367.97 = 449911.3 \text{ kWh} = 0.44991 \text{ GWh}$. This is 1.99% of MST's total energy consumption per year and almost twice as much as CT1.

2.3.3 Energy Consumption per Scan Type

Insight into the energy consumption overall and per scan type help provide arguments for working overtime or deliberately not working overtime. Energy consumption tells a lot about a scan. When a certain scan has higher energy usage than average, these scans can be evaluated to find causal relations between energy use and other factors. For example, there may be a causal relation between age or weight and energy use, since it is more difficult to insert an intravenous drip for contrast fluid in older and heavier patients. After examining energy consumption per scan type as a potential method for calculating scan costs and CO₂ emissions, we concluded that, due to the limited availability of data, only one week's worth, this approach was not feasible. However, given the value of this data for MST and future research, we have preserved it within our study. We needed our measured data and the performed time stamp per scan to calculate this. We have retrieved the latter from the program *Radimetrics*, where the actual start time of the scan is logged. We have used this to find the instant the scan started. For the end of the scan, we use one of the following assumptions:

- The beginning of the next scan.
- The moment the scanner returns to the idle state.

With the use of these assumptions and the retrieved data, Table 3 shows the average kWh per scan in a certain week in 2022.

Table 3: Scan types with average kWh used per scan from 18 to 25 October 2022 (MST, 2022)

Scan Type	Average kWh per scan
Abdomen_Art_Veneus	28.80
Abdomen_IVC	13.34
Abdomen_Lever_4Fasen	21.53
Abdomen_Micr_Hematurie	23.76
Abdomen_Nieren_Cyste	17.42
Abdomen_Pancreas_ART	14.05
Abdomen_RIP_Nieren	23.60
AbdomenIntervention	40.40
Aorta_Art_Veneus	10.03
Aorta_Arterieel	16.53
Aorta_Bekken_Benen	14.17
Brughoek	17.47
Buikvaten_DIEP	9.27
Cerebrum_Carotiden_Trombolyse	14.60
Cerebrum_Spiraal	23.78
Colografie_Blanco	22.33
Colografie_IVC	32.77
Hals_Bifasisch	18.60
Hals_Thorax_Abd_IVC	12.13
knie_prothese	4.27
Longembolie	17.60
Longembolie_Abd_Veneus	14.73
Longvenen_ECG	17.69
MST_Cor_Flash	19.56
MST_Cor_Seq_Auto	22.73
Petrosum	13.33
Sinus	4.73
Stereotaxie	13.27
TAVI	20.62
Thorax_Abdomen_IVC	11.53
Thorax_Blanco	5.87
Thorax_IVC	8.90
Thorax_Pancreas_ART	14.20

Figure 7 shows an example of an energy consumption course of a scan performed on the 19th of October 2022. We see the scan starting in the idle state of 44 kWh per minute rising twice to a peak between 132 and 176 kWh per minute during actual scanning and falling back to the idle state of 44 kWh per minute. This scan has consumed 22 kWh total.

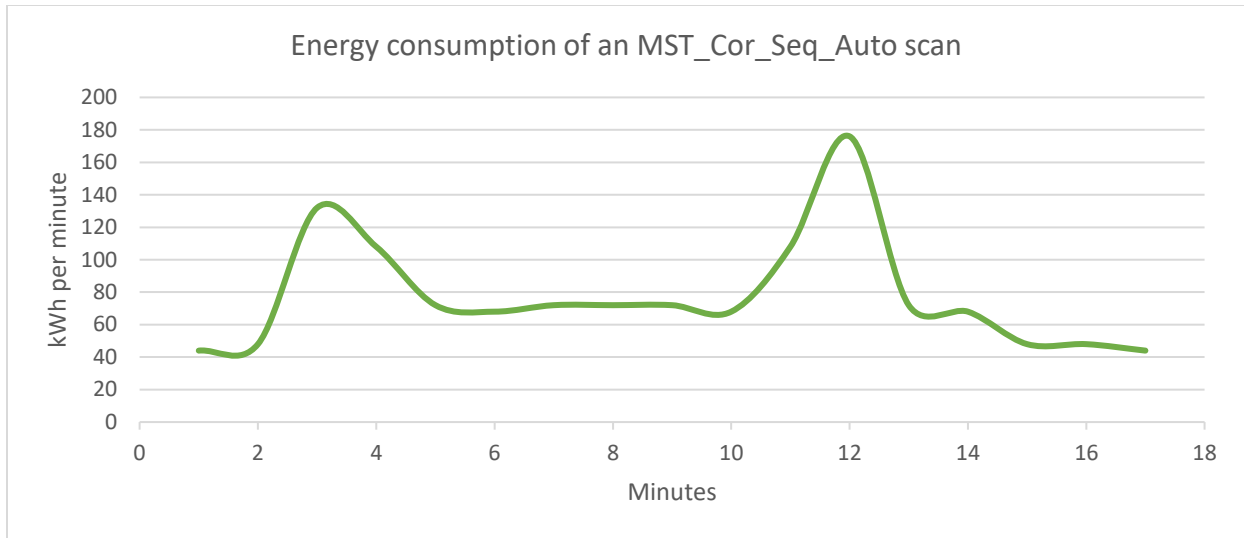


Figure 7: Energy consumption in kWh of an MST_Cor_Seq_Auto scan (MST, 2022)

2.4 CHALLENGES AND OPPORTUNITIES

The current CT schedule for MST is a difficult one to improve if we consider all of the complexities, but with a certain list of assumptions and simplifications, we scoped our research well and improved the schedule. It is complicated to model with multiple set requirements and variables to consider and also where there is limited to no data available about scan arrival rates and processing times. Section 4.3.3 shows later that most of the computed alternative values from historical data did not fit any common probability distribution, requiring more effort to process this data. One of the optimization opportunities lies in the data gathering along the way, since MST does not do this already. The data will add a different dimension to the current schedule evaluation and creates a new opportunity for unique insights that could not be seen previously.

We have highlighted the problem of the significant energy consumption in the healthcare sector, specifically in medical equipment such as CT scanners. However, there is a lack of knowledge regarding the energy usage of these appliances in MST. Our proposed approach to tackle this problem is to use the MPSM and its seven steps to systematically solve our problem using a Monte Carlo simulation and optimal job scheduling. We further elaborate upon the solution design in Section 4 of the research.

3 LITERATURE REVIEW

This section contains the literature review conducted to gain insights into what already has occurred in the field of CT appointment scheduling optimization concerning the radiology department or similar departments. Given the stochastic nature of scan processing times and scan arrival rates in our study, the utilization of Monte Carlo simulation presents promising opportunities for optimizing appointment scheduling. We have reviewed optimal job scheduling theory to define our problem environment. To get a better idea of key performance indicators, we sought fitting data analysis streams. Therefore, several streams of literature are relevant:

- Previous optimization techniques used for radiology scheduling or similar fields including heuristics.
- Monte Carlo simulation application in optimization of healthcare appointment scheduling.
- The definition of optimal job scheduling theory.
- Fitting qualitative and quantitative data analysis in our topic.

Section 3.1 handles the first two items. Section 3.2 continues with optimal job scheduling theory to gather knowledge on how to define our appointment scheduling problem. Section 3.3 holds literature on the qualitative and quantitative data analysis on classification, cluster, and cohort analysis to further examine the key performance indicators.

3.1 PREVIOUS LITERATURE ON APPOINTMENT SCHEDULING ACCOMPANIED BY SIMULATION

Abdalkareem et al. (2021) highlight the challenges of patient scheduling in healthcare systems, with emphasis on the impact of constraints such as personnel and resource limitations. They summarize recent studies about improving healthcare services and accessibility, while reducing costs, through the selection of hard and soft constraints. Hard constraints, such as room availability and necessary room properties, often cannot be violated, while soft constraints, such as preferred room properties and degree of specialization, are incorporated into the cost function to be minimized. In our case, we need to create assumptions to follow set constraints. We considered the hard constraints of machine availability and necessary properties for scans but did not consider the soft constraints such as the preferability of lab technician specialization, since those constraints do not apply to our study.

Koeleman & Koole (2012) conducted a study to examine appointment schedules while accounting for emergency arrivals that take precedence over scheduled appointments. To do so, they integrated these emergency arrivals into their appointment scheduling process. They then utilized a local search algorithm to identify the most optimal solution, which aimed to minimize the combined impact of overtime, idle time, and waiting times, weighted accordingly. We consider the integration of urgent arrivals in the scheduling process of normal scans on our emergency scanner and consider minimizing the impact on waiting times. We did not incorporate other factors such as overtime and weights into our study since the situation in their research differs from ours.

Borgman et al. (2018) discuss CT scheduling in a case study in a Dutch hospital and propose the use of a local search heuristic in combination with a simulation model to optimize the appointment schedule. Patient arrivals are decided first, secondly, patient selection for treatment based on prioritization is done and finally, non-treated patient urgency is updated. We did not consider patient urgency, since it is outside

of our scope, but have used the proof of the usefulness of heuristics and a simulation model in CT scheduling as a pillar for our choice of using a simulation model and heuristics.

Vermeulen et al. (2009) studied a CT-scan facility, including patient attributes such as urgency, request time, in/outpatient status, appointment duration, and contrast fluid needs. Their scheduling process, performed by administrative staff based on requests from lab technicians or radiologists, mirrors the characteristics of the current research. They use simulation to evaluate various scheduling methods and find that their adaptive allocation model, incorporating a rule for releasing reserved time slots for certain patients and using an algorithm to implement this into the scheduling system, outperforms the current system. This research also proves the usefulness of an algorithm in CT schedule optimization and therefore was another pillar for our approach of using simulation and heuristics.

Arnaout (2010) refers to heuristics for the maximization of operating room utilization, which is similar to CT scanner scheduling optimization, using simulation. *"[...] where most of the time an [examination] will not finish on a specific time, but on a range between two times. Simulation is considered to be one of the best approaches to deal with such a source of randomness."* (Arnaout, 2010, p. 578). Three heuristics are presented in this research: Longest Expected Processing Time (LEPT), Shortest Expected Processing Time (SEPT), and Longest Expected Processing with Setup Time (LEPST). He found LEPST to give the best results whereafter LEPT and SEPT subsequently followed. All heuristics give a certain priority to their respective expected processing time, whereas LEPST also includes the setup time. Expected processing time differs from processing time since it considers the average processing times of jobs. The advantage of using expected processing times is the ability to schedule jobs in stochastic environments. Our research does not consider setup times, so we considered LEPT and SEPT as valid heuristics to use in our study.

Lamiri et al. (2008) successfully prove the usefulness of Monte Carlo optimization for operating room planning with both elective and emergency patients in a stochastic environment, which is similar to our research environment. They mention their planning model is useful for hospitals using a 'blocked' advance scheduling system, which is the case in our study. They combine mixed integer programming into the Monte Carlo simulation to retrieve solutions to evaluate, which is a quite effective, yet straightforward way of simulating a problem environment and evaluating solutions tests. Therefore, our focus lied on using a Monte Carlo simulation for our research. We have described our problem environment as an optimal job scheduling problem as it was also proven to be successful as seen in the literature mentioned in this section.

3.2 MODELING APPROACH

The scheduling problem of this research draws many parallels with the scheduling theory also found in the articles from Section 3.1. Section 3.2.2 describes our scheduling problem, which is NP-hard (Pinedo, 2012). Heuristics are often used to tackle NP-hard scheduling problems since it is not feasible to find an optimal solution in a reasonable amount of time. Therefore, heuristics, which are algorithms that do not guarantee an optimal solution but aim to find a good solution within a reasonable amount of time, play an important role in solving NP-hard scheduling problems, and they are often the only practical approach to finding good solutions in a reasonable amount of time. This section elaborates on what a scheduling problem is and describes the theory. Section 4.2 discusses the scheduling problem of this research.

3.2.1 Scheduling

A scheduling problem deals with the allocation of tasks to resources over a set time period. Since these resources are generally limited, the tasks have to compete for the resources. The objective is to achieve an optimal schedule given certain criteria (Ruiz, 2015). To be more specific, scheduling is a decision-making process that is used often in various industries (Pinedo, 2012). These scheduling problems can be single- or multi-stage, where single-stage problems require only one singular phase and multi-stage problems require a task to pass multiple phases before completion. Multiple phases can take place in sequential or parallel. Next, we have examined the main machine environments (α), job characteristics (β), and objective functions (γ) as introduced by Graham et al. (1979).

3.2.1.1 Machine environments

3.2.1.1.1 Single-state job scheduling

- Single machine (1)
 - o Single-machine scheduling is the simplest case of machine environments where all possible jobs are scheduled on one machine.
- Identical machines in parallel (Pm)
 - o Identical machine scheduling has m identical machines in parallel. A certain job j only requires one operation which can be performed on any of the m machines.
- Machines in parallel with different speeds (Qm)
 - o Parallel machine scheduling also has m machines, but with different processing speeds denoted by v_i , meaning the speed of machine i . The time p_{ij} is the time that job j has spent on machine i and is calculated by dividing p_j by v_i if job j is fully processed on machine i . The machines are classified as uniform. If all machines have the same speed v , then the environment is equal to Pm .
- Unrelated machines in parallel (Rm)
 - o Unrelated parallel machine scheduling is a further generalization of the previous environment. With m different parallel machines, machine i processes job j at the speed v_{ij} . Time p_{ij} is the time that job j spends on machine i and is calculated by dividing p_j by v_{ij} if job j is fully processed on machine i . If the speed of all the machines is independent of the jobs, then the environment is equal to Qm .

3.2.1.1.2 Multi-stage job scheduling

- Flow shop (Fm)
 - o Flow shop scheduling has m machines in series. Each job will have to be processed on each of the machines. All jobs also have to follow the same route. The next job can only start when the last job has finished and after job completion on one machine, the job enters the queue on the next machine.
- Job shop (Jm)
 - o Job shop scheduling has m machines where each job has its predetermined route. Some jobs visit each machine only once and jobs may visit each machine more than only once.
- Open shop (Om)
 - o Open shop scheduling has m machines where each job has to be processed on each of the machines, but some processing times may be zero. The routing of the jobs has no

restrictions. The job schedule can be in any order and different jobs can have different routes.

3.2.1.2 Job characteristics

- Processing time (p_{ij})
 - The processing time of job j on machine i is the time a job takes to complete on that specific machine. If the processing time is not dependent on machine i or if the job is only processed on one specific machine, i is excluded.
- Release date (r_j)
 - The release date of job j is also known as the ready date. It is the moment the job arrives in the system, so the earliest moment the job can start its processing.
- Due date (d_j)
 - The due date of job j signifies the latest date a job should be completed. Completion after the due date is possible but is penalized.
- Weight (w_j)
 - The weight of a job can be seen as a priority factor where the importance of job j is noted and is relative to the other jobs in the system.

3.2.1.3 Objective functions

In scheduling problems, the goal is generally to minimize a certain objective value. All objectives can be summed with Σ or weighted with weight w_j per job. Several examples of objective functions are:

- Makespan (C_{max})
 - Makespan is equal to the completion time of the final job that leaves the system.
- Lateness (L_j)
 - Lateness measures the worst due date that has been violated.
- Flow time (F_j)
 - Flow time is the difference between the completion time and the release time.
- Throughput (U_j)
 - Throughput is the number of jobs that complete before their due date. This objective function is generally maximized instead of minimized.
- Tardiness (T_j)
 - Tardiness is positive lateness. If the lateness is negative, then the maximum tardiness is zero.
- Earliness (E_j)
 - Earliness is negative lateness and measures the difference between the due date and completion time. If the lateness is positive, then the maximum earliness is zero.

3.2.2 Our scheduling problem

Our scheduling problem consists of two identical CT machines in parallel where both machines can perform all jobs and all jobs only require one operation. As we aim to optimize the makespan of CT1 with the use of the processing time characteristic, we have described our scheduling problem as single-state, identical machines in parallel and denoted our problem as the following in the classical three-field notation of Graham et al. (1979):

$$P_2|p_j|C_{max}$$

Abdalkareem et al. (2021) state that in the last couple of years, the amount of healthcare scheduling methods that aim to optimize resource management in healthcare using heuristics has increased rapidly. They illustrate what large amounts of specific scheduling problems in healthcare have been solved separately and therefore show the potential of using scheduling theory to approach our scan scheduling problem. To translate our problem from theory into practice, we required to analyze our gathered data.

3.3 DATA ANALYSIS

To get a better understanding of how the CT scheduling and use process works, we needed to gather qualitative data. Next to that, we needed quantitative data to model a new, optimized schedule. We required to gather both types of data separately. When gathered, we required to process and analyze it by feeding it into a model to provide an improved or optimized schedule, because raw data is generally not prepared for straightforward understanding. This section elaborates on several ways of analyzing qualitative data and quantitative data.

3.3.1 Qualitative data

One of the goals of this research is to provide MST with a simulation model and schedule outcome to increase energy efficiency. This model is based on quantitative data. However, this quantitative information is pointless if you don't know where it comes from or what it implies. To have a deeper knowledge of the current situation, qualitative data is required. We have conducted numerous interviews to acquire qualitative data.

3.3.1.1 Interviews

To understand what data to gather and observe, we required to discover what data they gather, how certain choices in the scheduling process are made, and what constraints are in place in the current process. Appendix A shows an overview of interview questions. We then analyzed the obtained qualitative data using the Non-numerical Unstructured Data – Indexing, Searching, and Theorizing (NUDIST) method (Lillis, 1999). We incorporated topics for classification to produce a more comprehensive understanding of the offered solutions. Namey et al. (2008) suggest that qualitative data sets first need to be reduced. It is essential for identifying themes for data-driven analysis. The interviews gave a clearer view of how the process of CT scheduling works within MST and what quantitative data is already registered and more importantly, what quantitative data is not logged. Both insights aided the search and interpretation of current data, and what KPIs are within the system. Next to that, also where to focus on for the observational studies to gather non-existent data.

3.3.2 Quantitative data

As mentioned in Section 3.3.1, the expected deliverable of this research is based on quantitative data. This data needed to be gathered, combined, processed, and analyzed and that can be a time-consuming activity. A large amount of the gathered data consists of Excel files comprising tens of thousands of rows. These rows contain historical data on the scanning machines and patient appointments. The data should be the foundation of the research and show current performance within the radiology department. Due to the extent of the data, several methods were applied to create a practical overview of the necessary KPIs, namely scan processing time and total completion time.

3.3.2.1 Classification analysis

With the size of the gathered datasets, creating a structural overview could be difficult. Due to the several characteristics given per row, distinct variables could be identified. According to Tufféry (2011), classification is an operation that places data in specified classes, corresponding to certain characteristics of the data, which can be identified as independent variables.

With several large unconnected datasets containing unique information about the patients, the overlap of classifications could assist in the creation of a connection between these datasets. The overlap then could connect the classified objects, improving the overall structure. In the context of our research, we used patient A-numbers to link distinctive datasets and determine critical information such as scan type and time performed in one overview.

3.3.2.2 Cluster analysis

We then carried out a cluster analysis. Using similarities or differences, a clustering algorithm divides data objects into clusters (Frader & Matthiesen, 2010). These clusters have two properties (Tufféry, 2011). An example for this research would be the scan type and the number of total scans of the particular types. We then derived KPIs from the created cluster(s).

There are several types of clustering algorithms. Hierarchical clustering is an instance of non-parametric clustering (Fung, 2001) which in turn applies to the datasets within this research. A dendrogram is used for visualization of the algorithmic steps and will show which clusters are agglomerated in each step. We clustered data to find the number of scans with the same scan type.

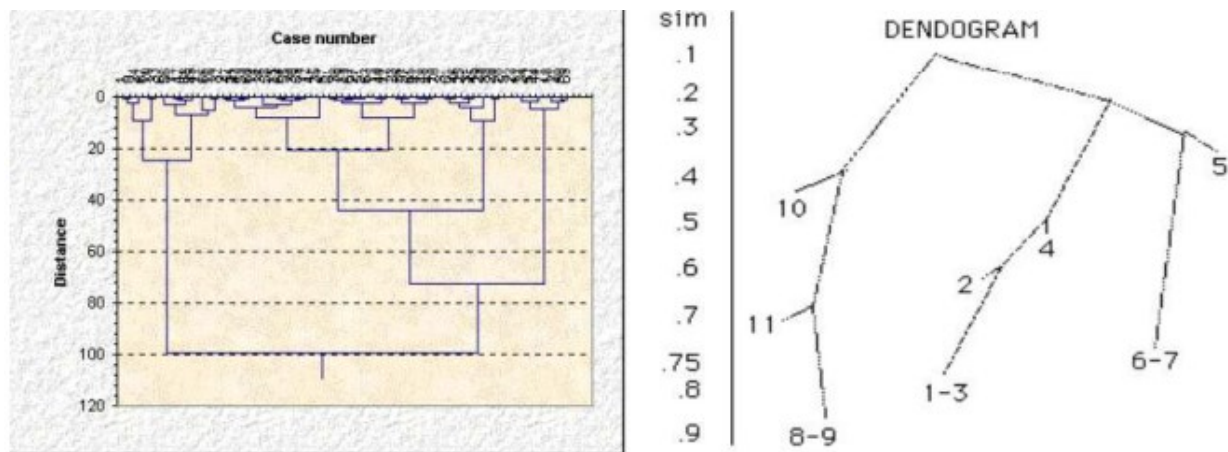


Figure 8: Dendrogram examples (Fung, 2001)

3.3.2.3 Difference between classification and cluster analysis

Although these two analysis methods look similar, there are some characteristic differences between these two algorithms. The classes are already defined in advance in *classification analysis* and its purpose is to label data, but the number of clusters is yet to be discovered during *cluster analysis* (Tufféry, 2011).

An example for clarification: during *classification analysis*, a patient is found with chosen characteristics. During *cluster analysis*, the number of patients with scan type X is determined.

3.3.2.4 Cohort analysis

A cohort is a group of objects entering a system at the same time. *Cohort analysis* is used to explain outcomes through the analysis of differences across two temporal dimensions: age and period (Mason &

Wolfinger, 2001). Cohort analysis is in other words an algorithm to analyze indicators over a time period. In this study, cohort analysis was used to seek how many scans and patient types are handled over a certain period. The data is able to provide hourly, daily, monthly, and yearly insights. This is to create for example an arrival distribution for emergency patients over a certain period of time.

4 SOLUTION DESIGN

In this section, we utilized the chosen methods of our literature research to create workable data and prepare for the simulation. As mentioned in 1.4, we have chosen a Monte Carlo simulation as a useful method for dealing with our stochastic elements such as the probabilistic scan duration, and it is simple to implement. Section 4.1 elaborates upon the quantitative data where Section 4.2 continues describing the scheduling problem and the proposed heuristics. Section 4.3 describes the entire Monte Carlo simulation with the necessary inputs.

4.1 QUANTITATIVE DATA ANALYSIS

We have gathered a large amount of raw quantitative data that we had to prepare to utilize in our simulation. Also, the translation from qualitative data to KPIs gave a good foundation for quantitative data analysis. The different analyses that we have performed in this section are classification and cluster analysis. We performed these to organize the data for usage as simulation input.

4.1.1 The observational study

The interviews of the lab technicians and the observations done at CT1 and CTSEH gave a clear overview of what data was available and what was not available. Important performance indicators retrieved from the datasets are:

- Scan type
- Scan performed time stamp
- Planned scan duration

Crucial information about scan duration was not readily available and we attempted to retrieve it by joining lab technicians at CTSEH for two days and logging data manually amongst others:

- Patient urgency
- Patient arrival
- Patient departure

We retrieved 97 data points during this time period and several insights into how patient urgency is appointed, appointment slots are filled, and the general order of business. The data gives insights into the validity of similar data that we have computed instead of logged.

4.1.2 Assumptions

To translate the datasets into simulation input, we made assumptions. Performed time stamp per scan type could not immediately give insight into scan duration. To calculate the average scan duration per scan type, we have subtracted the performed time stamp of a certain scan type from the performed scan time of the consecutive scan. We excluded scans outside of general operation hours and scans during the general lunch period from calculations since there was no guarantee that the next scan was scheduled immediately after the current scan. Other assumptions include:

1. All patients arrive on time for their appointment.
2. All emergency scans are performed immediately or as soon as possible, i.e., scheduled at arrival time.

3. All emergency patients have the same level of urgency.
4. CT1 can be shut off after all scheduled scans are performed.

4.1.3 Classification analysis

Next, we conducted a classification analysis to better comprehend the data acquired for this study. The segmentation of the data into smaller groups is made easier by the classification. The performed analysis groups the data into connected objects based on certain characteristics, which in turn can be identified as independent variables (Tufféry, 2011). The classification of the objects and the created relationships between data objects gave a clearer view of the KPIs and the simulation model input.

Remembering our important performance indicators, the classifications became more obvious. Some variables are also independent variables of the data objects. An example hereof is a patient. It has characteristics that are unique for each patient and aid the data analysis for the simulation input such as the arrival rate. Fitting independent variables are their CT processing time and arrival time.

We fitted patients in the following classes and objects:

- Patient
- Scan
 - SPS (C) Code
 - Scan duration
 - Scan type
 - Scan module

From the 42 different variables, we have only used the six independent variables from the list above. We have removed most of the variables due to them not adding value to our research goal and simulation model.

4.1.4 Cluster analysis

To evaluate simulation input, we required a cluster analysis. We formed clusters from created classes and independent variables to analyze and process the necessary KPIs. By adding criteria to the clusters, we retrieved more specific results to evaluate the simulation input.

Clusters that we identified and are of potential use for the simulation or further research are whether patients arrive on time or the no-shows per scan type. Clusters like these aided in gathering more realistic data to use in a more detailed simulation.

4.1.5 Cohort analysis

We observed the created clusters from the cluster analysis during a certain time period to perform the cohort analysis. The cohort analysis helps to find changes in outcomes of certain KPIs over time to for example find the seasonality. Examples of cohorts are processing times and arrival rates during different months of different scan types per day or hour to potentially compute the arrival rates per time period.

4.2 SCHEDULING PROBLEM & HEURISTIC

In this section, we describe the scheduling problem of CT scanners within the radiology department of MST. We also propose and elaborate on two heuristics for CT scan scheduling.

4.2.1 CT Scheduling Problem

MST intends to become more sustainable and therefore tries to focus more on working more sustainably. In this specific case, they aim to reduce the maximum completion time of the CT scanners in the radiology department to increase the energy efficiency of the CT scanners. Reducing the maximum completion time of the scanners with the same number of scans reduces the idle time between scanning. This means we can complete the same number of scans in less time than previously necessary and increase the useful energy usage of the equipment. When this is realized, less energy is wasted, and energy efficiency is increased. While aiming to be more sustainable, the healthcare service level may not decrease substantially. In scheduling terms, the maximum completion time is noted as C_{max} and is also known as the makespan. Each scan, or job, comprises only a single execution before completion and therefore this scheduling problem is a single-state, identical parallel machine scheduling problem.

Our scheduling problem consists of online, offline, deterministic, and stochastic elements. In our simulation, all normal scans are known before scheduling and therefore consist of offline planning. On the other hand, all urgent scans are scheduled as they enter the system and are thus scheduled through online scheduling. The patient arrival rate is a Poisson process and is a stochastic element, but the expected processing times are fixed and hence have a deterministic element. Ultimately, to evaluate the schedule, we generate the processing times for the 20 selected scan types and examine the outcome of the schedule. This means that our scheduling problem is stochastic.

We have executed the simulation for CT1 and CTSEH. Our scheduling problem is nondeterministic polynomial time (NP) hard (Pinedo, 2012). This means that this problem is widely considered one of the most challenging issues to find an optimal solution for.

Since the scheduling problem involves CT1 and CTSEH, we have used the processing time as input for the simulation. We aimed to minimize C_{max} and modeled our scheduling problem with two machines incorporating the processing time of the jobs to reach our goal. We have described the scheduling problem as the following, according to literature from Section 3.2.1 and Graham et al. (1979):

$$P_2|p_j|C_{max}$$

4.2.2 Heuristics

Several heuristics were found in Section 3.1 from Arnaout (2010), such as Longest-Expected-Processing Time-first (LEPT), Shortest-Expected-Processing-Time-first (SEPT), and Longest-Expected-Processing-and Setup-Time-first (LEPST). These are well-known heuristics that are commonly applied to many scheduling problems that aim to minimize makespan based on processing time. He introduces the variations LEPT and SEPT as an addition to Longest Processing Time-first (LPT) and Shortest-Processing-Time-first (SPT) respectively. The LPT heuristic gives priority to the job with the longest processing time when scheduling, which means that the shortest job will be the last to be assigned to a machine. On the other hand, the SPT heuristic gives priority to the job with the shortest processing time when scheduling. As a result, the longest job will be the last to be assigned to a machine.

Where LEPT and SEPT differ from LPT and SPT respectively is that LEPT and SEPT not only evaluate data based on processing time but consider average processing time. Due to this, these heuristics apply to scheduling problems in a stochastic environment. Arnaout evaluated his stochastic scheduling problem using LEPST, LEPT, and SEPT to maximize operating room utilization and minimize makespan including setup times. He found that LEPST was the superior heuristic, followed by LEPT and SEPT. Since our

research scope does not include setup times, we did not use the LEPST heuristic but selected the LEPT and SEPT heuristics to evaluate our scheduling problem during this research and evaluate both heuristics.

Since heuristics function as approximation algorithms, it is important to consider their performance and worst-case ratios in comparison to an optimal solution. Graham (1966) conducted an analysis of worst-case scenarios for list scheduling policies, including LPT and SPT, and determined a worst-case ratio of $2 - \frac{1}{n}$ where n represents the number of machines. This indicates that the LPT and SPT heuristics yield outcomes that are at most $2 - \frac{1}{n}$ times worse than an optimal solution. Additionally, Pinedo (2012) has provided evidence concerning multiple machine scheduling problems without preemptions in a deterministic setting, demonstrating that the worst-case ratio for minimizing makespan is $\frac{4}{3} - \frac{1}{m}$ with m denoting the number of machines involved.

In a different context, Sagnol et al. (2018) demonstrated that LEPT, when applied to fixed surgery job assignment, exhibits a performance ratio of $1 + e^{-1}$ and that this “[...] shows that [LEPT] is –in a certain sense– the best possible fixed assignment policy for a natural assumption on the processing time distribution.” (p. 334). However, it is crucial to note that each heuristic may yield different ratios depending on various assumptions and scheduling problem types. To the best of my knowledge, there has been no research done on the exact worst-case ratio of the scheduling problem noted in this research. Nevertheless, considering the promising performance ratio reported by Sagnol et al. (2018), it influenced our decision to utilize LEPT in our research.

4.2.2.1 LEPT

We defined the LEPT heuristic and the three steps according to Arnaout (2010). LEPT picks the job j from the set of unscheduled jobs with the maximum expected processing time EP_j and assigns it to machine i . $C_{k,i}$ is the completion time of the previous scan k on machine i . We add the constraint that job j cannot use more than 3 (5-minute) scheduling slots on machine CTSEH. This assumption is based on what we have observed in the data.

Let (S) be a set containing the unscheduled scans:

1. Identify job $j \in (S)$ that has the **maximum** value for (1). If the machine is CTSEH, then EP_j should be equal to or lower than 3 ($EP_j \leq 3$).

$$C_{k,i} + EP_j \tag{1}$$

2. Place job j on machine i and remove the job from set S .
3. If set S is empty ($S = \emptyset$), then stop. Else, return to step 1.

4.2.2.2 SEPT

SEPT is also defined using three steps according to Arnaout (2010). SEPT picks the job j from the set of unscheduled jobs with the minimum expected processing time EP_i and assigns it to machine i . $C_{k,i}$ is the completion time of the previous scan k on machine i . Here we also add the constraint that job j cannot use more than 3 (5-minute) scheduling slots on machine CTSEH. This assumption is based on what we have observed in the data.

Let (S) be a set containing the unscheduled scans:

1. Identify job $j \in (S)$ that has the **minimum** value for (1). If the machine is CTSEH, then EP_j should be equal to or lower than 3 ($EP_j \leq 3$).

$$C_{k,i} + EP_j \tag{1}$$

2. Place job j on machine i and remove the job from set S .
3. If set S is empty ($S = \emptyset$), then stop. Else, return to step 1.

4.3 MONTE CARLO SIMULATION

To evaluate the proposed heuristic, we have designed and run a Monte Carlo simulation. This section describes the data preparation for the simulation input and how we created and performed the simulation.

4.3.1 Data preparation

The datasets used in this research are from 2019, 2020, and 2021. Since COVID-19 has influenced data from 2020 and 2021, we used the historical data from 2019. The data, shown in an Excel sheet, shows most importantly scan types and appointment times, but unfortunately not the scan duration or scan urgency. The scan duration is derived by subtracting the two consecutive appointment times and is evaluated by comparing the estimated scan duration with the planned scan duration. We removed data values near lunchtime or at the end of a day from this data analysis since a lunch break and the last scan of the day would not give valid results in this calculation. Operating hours were set between 08:00 and 16:30 during weekdays. Lunchbreak would end at 13:00, so we secluded the scans where the following scan would start at 13:00. For CTSEH, we used the scheduled scan duration.

4.3.2 Data analysis

During the data analysis for the simulation input, we calculated the percentage of individual scan types of the total of scan types per year. In 2019, there were a total of 102 different scan types that have been carried out on CT1 and 107 on CTSEH. To keep within the scope of the research, we did calculations over the scan types on CT1 with an occurrence rate of over 1% of the total. There are 20 scans with an occurrence rate above the 1% boundary. The remaining 82 scan types had such a low occurrence rate that we did not use their data for calculations since they would not be accurate enough to use during this research. Appendix E.1 shows all scan types of CT1 and CTSEH and their occurrence rate.

To retrieve the probability distributions of the different scan type processing times, a chi-squared test was performed for each of the 20 scan types to compare expected distributional results with the observed results for comparison. We first used Descriptive Statistics and later use the Histogram function within Excel. Then we computed α , β and the number of bins. We computed the number of bins by calculating \sqrt{n} , whereafter the bins with fewer than 5 observations were combined with the previous bin (Field, 2009). Next, we calculated the Cumulative Distribution Function (CDF) using Excel functions for the following distributions: Exponential, Gamma, Lognormal, Normal, and Weibull. We then multiplied the CDF of every separate bin by the observation count whereafter we calculated the number of expected observations per bin. We compared the expected observations with the actual observations using the following formula:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

χ^2 = chi-squared, O_i = observed value, E_i = expected value

We computed the total chi-square by taking the sum of all separate chi-square per bin and the test hypothesis variable. We retrieved the critical value by performing the Excel function for the right-tailed chi-squared probability function. The input for this chi-square inverse root function is the probability, also known as the significance level, and the degrees of freedom. The significance level was set at 0.05 and the degrees of freedom equals the number of bins minus one. The last step of the evaluation is then to compare the results of the chi-square and critical value. When the chi-square value is lower than the critical value, we do not reject the null hypothesis that the statistical distribution fits the respective dataset. When the chi-square value exceeds the critical value, we reject the null hypothesis, and the statistical distribution does not fit the respective dataset.

Unfortunately, only 4 of our 20 chosen scan types' processing times of CT1 fitted a statistical distribution. Table 4 displays these scan types and their respective distributions.

Table 4: 20 chosen scan types and their distributions.

Scan type	Distribution
C7-71E	Empirical
C7-71D	Empirical
C5-57	Empirical
C1-20	Empirical
C5-59	Empirical
C5-50	Empirical
C7-70G	Empirical
C2-21	Gamma
C7-72A	Empirical
C5-61	Empirical
C5-59A	Empirical
C7-70I	Empirical
C7-90C	Empirical
C7-70N	Empirical
C5-50	Exponential
C1-01	Exponential
C7-74	Empirical
C7-70J	Empirical
C2-24	Exponential
C7-75	Empirical

Since we could reject no other null hypothesis using the chi-square test on the aforementioned statistical distributions, the scan types' processing times were assessed using empirical distributions. Empirical in this context means a derived distribution by calculation using the observed frequency of scan durations.

4.3.3 Simulation Input

For the actual simulation, several variables are necessary as input:

- Scan processing time
- Scan arrival time
- Number of iterations

4.3.3.1 Processing times

We computed the processing times of the chosen scan types with an empirical distribution by executing several steps. We divided the number of observations per bin type by the total bin type to get the probability of scan processing time per bin type. Next, we computed the cumulative probabilities by addition. Since the total cumulative probability lies between 0 and 1, we could now generate processing times per scan with the use of a random number between 0 and 1. By generating a random number between 0 and 1, we compared the random number with the lower and upper bounds of every bin. Once we found a match, we generated a random number that lies within the selected bin. To compute the processing times of the gamma and exponentially distributed scan types, we generated a random variable from their respective distribution. For the gamma distribution we have used the Gamma Inverse Excel function and for the exponential function we used the following formula:

$$E = -LN(1 - X)/\lambda$$

E = random generated exponential value, X = random generated value between 0 and 1, $\lambda = 1/\text{mean}$

We stored all generated processing times in separate sheets per scan type but did not indicate them in the Monte Carlo Simulation sheet, since the generated processing time per scan is no input that should be modified and therefore is stored somewhere else. For the processing times of the remaining 82 scan types of CT1 and the 107 scan types on CTSEH, we chose the respective processing time that MST has scheduled for those scan types during the scan generation process. We have retrieved these values from the scheduling provided MST data. Appendix D.1 shows the Excel VBA code that we use for these calculations. The green text within the code is called pseudo code. Pseudo code is a comment that clarifies the lines of code following or preceding it.

We gave every scan type a scheduled processing time, also known as scan duration. This is the amount of time a scan is given before the next scan should start, meaning when a patient enters the room until he leaves the room. Unfortunately, the actual processing time was not logged, so we took an alternative approach to conceiving this data.

Table 5 shows the calculated average processing time of the 20 chosen scans where the computed mean processing time values are in seconds. The calculated processing times could be relatively high in comparison with the actual processing times due to a gap in the schedule since such a condition was difficult to implement in the analysis. For our calculations, we used the generated values since these values reflect the actual situation.

Table 5: Scheduled and actual processing time per scan type on CT1 in seconds in 2019 (MST, 2022)

Scan Types	Scheduled time	Calculated average processing time
C7-71E	900	1012.61
C7-71D	900	1039.43
C5-57	900	1060.30
C1-20	900	1121.02
C5-59	1800	1749.97
C5-50	900	832.67
C7-70G	900	1009.57
C2-21	300	627.05
C7-72A	300	992.50
C5-61	900	1178.10
C5-59A	900	1061.05
C7-70I	900	1007.31
C7-90C	1800	1826.09
C7-70N	900	1011.99
C5-50C	300	672.79
C1-01	300	890.89
C7-74	300	982.31
C7-70J	900	1000.00
C2-24	300	610.54
C7-75	300	1182.86

4.3.3.2 Scan Arrival Rate

Scans arrive at the scheduling system with a specific arrival rate and are scheduled accordingly. The number of scans that are requested per time period was not logged unfortunately and logging it manually was difficult, since multiple streams allow a scan to be scheduled as mentioned in Section 2.2.5. Therefore, we used Hofman (2014) to use accurate historical data that resembles our environment. In his work for a similar radiology department, he defined the echography arrival rate as the scan request rate for different patient types. We used these rates as a basis for our arrival factors. For our simulation of the CT1 and CTSEH, we combined arrival rates into two rates: normal and urgent. We adjusted these rates into arrival rate factors relative to each other within their own column where the sum of these hourly, daily, or weekly factors is 1. To calculate the respective hourly and daily arrival rates in our simulation, we used the average amount of scans in 2019 retrieved from historical data and multiplied it with each respective factor to get the following arrival rate:

$$\lambda_{d,h}$$

d = day (Monday, Tuesday, Wednesday, Thursday, Friday), h = hour (8, 9, 10, 11, 12, 13, 14, 15, 16)

Table 6 and Table 7 show the calculated arrival rate factors. We assumed that all patients arrive at the beginning of their arrival hour.

Table 6: Arrival rate factors for normal-type patients in the number of patients per hour per day

Day Hour	Monday	Tuesday	Wednesday	Thursday	Friday
8	4.68	4.72	4.75	4.79	4.89
9	3.91	3.95	3.97	4.01	4.09
10	4.23	4.28	4.30	4.34	4.43
11	3.08	3.11	3.13	3.16	3.22
12	0.97	0.98	0.98	0.99	1.01
13	2.72	2.74	2.76	2.79	2.84
14	2.53	2.55	2.57	2.59	2.64
15	3.31	3.34	3.36	3.39	3.46
16	1.26	1.27	1.28	1.29	1.32

Table 7: Arrival rate factors for urgent-type patients in the number of patients per hour per day

Day Hour	Monday	Tuesday	Wednesday	Thursday	Friday
8	8.53	7.40	5.95	8.45	9.90
9	7.15	6.20	4.99	7.08	8.29
10	7.72	6.70	5.39	7.65	8.96
11	5.65	4.90	3.94	5.60	6.55
12	1.73	1.50	1.21	1.71	2.01
13	4.96	4.30	3.46	4.91	5.75
14	4.61	4.00	3.22	4.57	5.35
15	6.11	5.30	4.27	6.05	7.09
16	2.31	2.00	1.61	2.28	2.68

4.3.4 Scan Generation

The next step in the simulation is the generation of scans to schedule. The simulation had been set to generate a single working week of scans per replication. The mean number of scans conducted on CT1 in 2019 was determined to be 133, with a standard deviation of 18.9. The mean number of scans conducted on CTSEH during CT1 working hours in 2019 was determined to be 230, with a standard deviation of 29.7. We calculate this through the following equations:

$$\frac{\text{total amount of scans in 2019 on CT1 between 08:00 and 16:30 on weekdays}}{\text{number of working days}} * \text{five working days} = \frac{6951}{261} * 5 \approx 133$$

$$\frac{\text{total amount of scans in 2019 on CTSEH between 08:00 and 16:30 on weekdays}}{\text{number of working days}} * \text{five working days} = \frac{11968}{261} * 5 \approx 230$$

We generated the scans in the same way as the scan processing times. We calculated the occurrence rate of the separate scan types and the cumulative probability. Next, we also generated a random number between 0 and 1 and compared it with the lower and upper bounds of each scan type's cumulative probability. The simulation then determined if the scan type is part of the 20 chosen normal scan types or

the other 82. If it is part of the 20 chosen normal scan types, it randomly selects a value from the generated processing times as processing time. If it is part of the other 82 scan types or an urgent scan, it selects the scheduled time from the MST data as the processing time for the respective scan type. Appendix D.2 shows the Excel VBA code that we have used for these calculations.

4.3.5 Reservation slots

To optimize the CT1 schedule, we have proposed a strategy to reschedule normal scans that are currently scheduled on CT1 to instead take place on CTSEH. Normal scans cannot be randomly scheduled on CTSEH and should have predetermined slots where they can be scheduled in. Scheduling scans on CTSEH may result in increased waiting time for urgent patients. To diminish this issue, we have proposed the allocation of a reserved set of slots for normal scans on CTSEH. Our proposed approach aims to minimize increases in waiting time for urgent patients while also reducing energy consumption on CT1. To establish a baseline for urgent patient waiting times, we have performed simulations without reserved scheduling blocks. We determined the required number of replications to achieve reliable waiting time results at a 95% confidence interval via calculations, as detailed in Appendix E.1. We attained the accuracy of waiting time values after 374 replications, and we utilized the average percentage of empty CTSEH scheduling blocks to determine the necessary blocks to reserve for normal scans.

We did the following to compute a proposed reservation slot schedule. First, we converted the average percentage of empty CTSEH scheduling blocks to rates relative to each other per 5-minute scheduling block over 5 days, with the sum equating to 1. Secondly, we multiplied the average percentage of empty blocks per hour by the number of blocks per hour. We subsequently multiplied the average of empty slot values per hour per day by the average total empty blocks per hour per week. Figure 9 shows the resulting values with the table displaying the number of blocks reserved at the end of each hour per day. For example, during hour 8 on day 1, the last two blocks at 08:50 and 08:55 are reserved in line with this schedule. Appendix E.2 gives more information on the calculation and the numbers.

Reserved slots per hour per day					
	1	2	3	4	5
8	2	2	2	2	2
9	2	2	2	2	2
10	2	2	2	2	2
11	3	3	3	3	3
12	7	7	8	7	7
13	5	5	5	5	5
14	5	5	5	5	5
15	4	3	4	3	4
16	2	2	2	2	2

Figure 9: Reservation slots per hour. The horizontal label shows days 1 to 5, the vertical label shows hours 8 to 16.

4.3.6 Scan Scheduling

Next, we have applied the proposed scheduling heuristic in this part of the simulation to evaluate the influence of the heuristic on the scheduling system. The urgent scans are performed as soon as possible after entering the system and hence put in the schedule immediately in the first non-reserved block. We have generated the normal scans to schedule them during the simulated week and therefore already have them in our theoretical system before the Monday that the first scan is scheduled. The proposed LEPT and SEPT heuristics search for the normal job with the longest and shortest expected processing time respectively to schedule it first in a weekly schedule. We have ordered the normal scans descending and

ascending respectively in their array, whereafter the code only must loop over all generated scans instead of searching for the scan with the highest or shortest processing time. Scans are scheduled from 08:00 to 16:20. The 5-minute scheduling blocks are then filled with the scans based on their scheduling time either on CT1 or CTSEH, depending on if there are enough reserved blocks on CTSEH and the processing time of the normal scan. Since we do not prefer to interrupt a scan after it has started, we only allow scans that take 3 or fewer scheduling blocks to be scheduled in the CTSEH schedule. When the schedule is filled for the day, the scan is planned for the next day. Additionally, when a scan arrives at the scheduling code, the code loops over every timeslot from the moment the scan arrives at the system starting on CTSEH. Then, when the block is reserved on CTSEH and not filled yet, the code looks if enough consecutive slots are available to schedule the scan. If that is not the case, the code searches if the same block is available in the CT1 schedule. If that also is not the case, the code looks in the next timeslot starting in the CTSEH schedule again. For the LEPT heuristic, the code runs starting with the longest scan and for the SEPT heuristic, the code starts with the shortest scan. Appendix D.3 shows the Excel VBA code that we use for these calculations.

When an urgent patient arrives at the hospital and a normal scan is already scheduled in a reserved block, the simulation would schedule the urgent patient in the earliest block after the normal patient. In reality, depending on the urgency, the urgent patient would be scanned before the normal patient and the normal patient would be scanned once the urgent scan is finished. This influences the maximum completion time at a minimum since nearly all reserved blocks are filled with normal scans, and the scans would simply be switched and still have the same combined completion time.

4.3.7 Replications

Monte Carlo simulations are a method for estimating the results of stochastic events by repeatedly predicting outcomes (Harrison, 2010). To improve the accuracy of the results, we must replicate the scan generation and scheduling multiple times (Mundform, et al., 2011). To calculate how many times, we evaluated our maximum completion time to find the minimum number of replications with a confidence interval of 95%, which is after 19 replications. We have added a few replications for certain accuracy and use 25 replications. Appendix E.1 shows these calculations. We retrieved the number of replications for which the simulation iterates from the 'Monte Carlo Simulation' Excel sheet input values and integrate it into the simulation's code. The simulation outputs the average maximum makespan per day, which is the KPI we have assessed to evaluate our chosen heuristics. Appendix D.4 shows the Excel VBA code that we use for these calculations.

4.3.8 User Display

The Monte Carlo simulation has a user display where input variables are subject to change, output variables are found, and the simulation outcome is seen. Figure 10 shows this user display. We do not show all rows in the figure because displaying all rows would decrease the visibility of the user display. The simulation input and output are housed within the left dark green blocks labeled 'Input' and 'Output' respectively. These blocks serve as containers for the relevant data and results of the simulation, with the number of replications being the adjustable variable. Figure 9 shows the only user input that is not found in Figure 10. The arrival factors for each time period and the reserved timeslots are potentially modifiable but are maintained unless more suitable variables are discovered. The variation in arrival rate or reserved timeslots can be tested through the modification of these variables. The output of the simulation is shown under the 'Output' label and consists of the average maximum makespan per working week, which is

compared with the current situation in Sections 5 and 6. Appendix F shows the output per replication and its respective reserved timeslot schedule.

The ‘*Generated Normal Scans*’ and ‘*Generated Urgent Scans*’ labels contain the normal (N#) and urgent (U#) scans generated for a typical week, which are subsequently scheduled using our heuristics. The simulation output includes scan details, such as scan numbers, types, processing time, arrival day, and arrival hour, and the resulting schedule output generated via the LEPT and SEPT heuristics. The schedule output displays the number of scans scheduled per day and their respective total completion time. In addition, the simulation timer is located on the far right of the display.

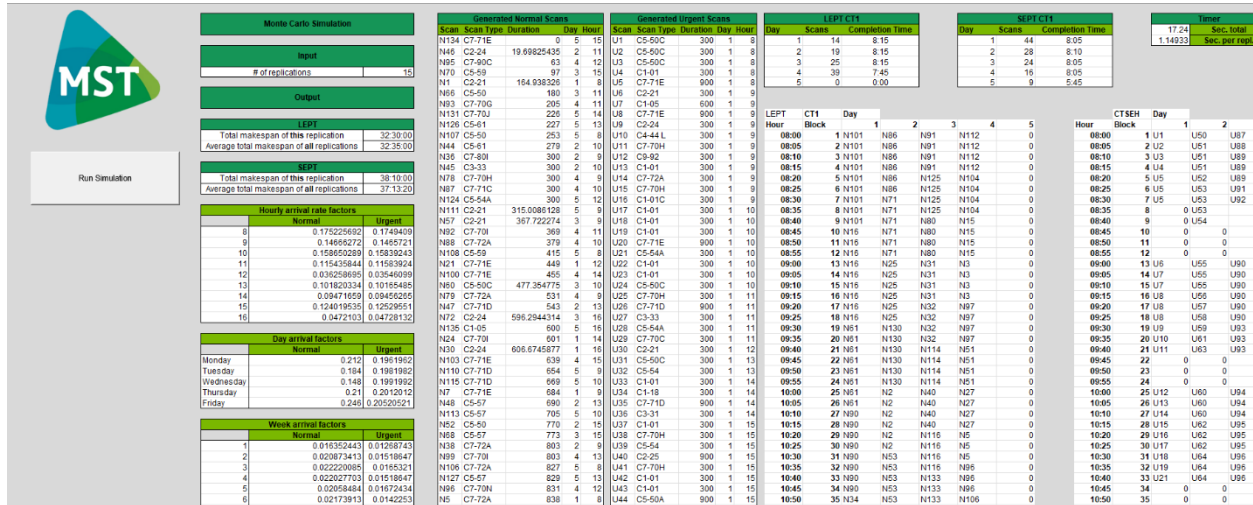


Figure 10: User Display of the Monte Carlo simulation

We see several schedules in the dashboard as well. The simulation creates 5 schedules for every experiment in the dashboard:

- LEPT
 - CT1
 - CTSEH
- SEPT
 - CT1
 - CTSEH
- CTSEH without normal scans

The schedules have 5-minute scheduling blocks for every workweek from day 1 to 5 for scans to be scheduled. Figure 10 and Figure 11 do not show all schedules, but they all are visually similar. All this data is generated when the Monte Carlo simulation is started with the ‘Run Simulation’ button.

The simulation runs for the number of replications that are input. For accurate results, we have chosen to use 25 replications to generate scan arrivals and processing times for scans based on the stochastic distributions mentioned in Section 4.3.7. The simulation took around 181 seconds to run and averaged 7 seconds per replication.

4.3.9 Schedule outcome

One can argue that relying solely on either expected or generated processing times for scheduling and results may not yield the most accurate outcomes. The expected processing times are the predefined times used by MST to schedule different scan types within their regular operations. These times are based on their experience and knowledge of the scanning process. On the other hand, the generated processing times are derived from historical data and represent the stochastic nature of the scanning process. These generated processing times capture the variability and uncertainty associated with each of the 20 scan-type processing times.

To address this issue, we generated the stochastic processing times for a selected set of 20 scan types in each of the 25 replications, as described in Sections 4.3.2 and 4.3.7, to generate the outcome of the schedules. By incorporating both expected processing times and generated processing times in our simulation, we could evaluate the performance of different scheduling strategies under realistic conditions. This approach allows us to assess the effectiveness of the scheduling system while incorporating variations in processing times.

Section 5 contains the simulation results, where we find the scheduled total makespan and the total makespan of the outcome of the schedule. Figure 11 shows an example of a schedule versus its outcome. The outcome is generated using generated processing times for selected normal scans and expected processing times for all other scans. If an urgent scan has already arrived at the hospital and there are empty reserved scheduling blocks, the patient can be treated earlier than scheduled. The average total makespan of all replications (Avg. tot. C_{max}) is calculated for each respective schedule.

The left cluster under the label '*Scheduled*' displays the schedule for a certain week generated using the LEPT heuristic for CT1. The right cluster under the label '*Outcome*' shows the fulfilled schedule. The 'Avg. tot. C_{max} ' is the average total makespan of all replications, and in this example, the 26 hours and 42 minutes for CT1 LEPT scheduled is the average total makespan of the 25 replications performed for this result. In case a scan takes more time than expected the schedule for that day shifts the amount of blocks the scan took extra. In case a scan takes less time than expected, the following scan can only start on or after its scheduled block, as patients' early or late arrival is not considered in this study. As an example, we can see that N25 takes one scheduling block less than expected, but N35 three blocks more.

Scheduled		Avg. tot. Cmax		26:42:48		Outcome		Avg. tot. Cmax		33:01:24			
LEPT	CT1	Day					LEPT	CT1	Day				
Hour	Block	1	2	3	4	5	Hour	Block	1	2	3	4	5
08:00	1 N16	N15	N65	N127		0	08:00	1 N16	N15	N65		0	0
08:05	2 N16	N15	N65	N127		0	08:05	2 N16	N15	N65		0	0
08:10	3 N16	N15	N65	N127		0	08:10	3 N16	N15	N65		0	0
08:15	4 N16	N11	N68	N126		0	08:15	4 N16	N15	N65	N126		0
08:20	5 N16	N11	N68	N126		0	08:20	5 N16	N11	N68	N126		0
08:25	6 N16	N11	N68	N126		0	08:25	6 N16	N11	N68	N126		0
08:30	7 N25	N21	N69	N129		0	08:30	7 N25	N11	N68	N126		0
08:35	8 N25	N21	N69	N129		0	08:35	8 N25	N11	N69	N129		0
08:40	9 N25	N21	N69	N129		0	08:40	9 N25	N21	N69	N129		0
08:45	10 N25	N18	N72	N128		0	08:45	10 N25	N21	N69	N129		0
08:50	11 N25	N18	N72	N128		0	08:50	11 N25	N21	N69	N128		0
08:55	12 N25	N18	N72	N128		0	08:55	12	0 N21	N72	N128		0
09:00	13 N29	N19	N76	N132		0	09:00	13 N29	N18	N72	N128		0
09:05	14 N29	N19	N76	N132		0	09:05	14 N29	N19	N72	N128		0
09:10	15 N29	N19	N76	N132		0	09:10	15 N29	N19	N76	N132		0
09:15	16 N29	N20	N77	N133		0	09:15	16 N29	N19	N76	N132		0
09:20	17 N29	N20	N77	N133		0	09:20	17 N29	N19	N76	N133		0
09:25	18 N29	N20	N77	N133		0	09:25	18	0 N20	N77	N133		0
09:30	19 N35	N23	N70	N134		0	09:30	19 N35	N20	N77	N133		0
09:35	20 N35	N23	N70	N134		0	09:35	20 N35	N20	N77	N133		0
09:40	21 N35	N23	N70	N134		0	09:40	21 N35	N20	N77	N133		0
09:45	22 N35	N22	N71	N131		0	09:45	22 N35	N23	N77	N133		0
09:50	23 N35	N22	N71	N131		0	09:50	23 N35	N23	N77	N133		0
09:55	24 N35	N22	N71	N131		0	09:55	24 N35	N23	N77	N133		0
10:00	25 N63	N24	N82	N136		0	10:00	25 N35	N22	N77	N133		0
10:05	26 N63	N24	N82	N136		0	10:05	26 N35	N22	N77	N133		0
10:10	27 N63	N24	N82	N136		0	10:10	27 N35	N22	N70	N133		0
10:15	28 N63	N26	N83	N135		0	10:15	28 N63	N22	N70	N133		0
10:20	29 N63	N26	N83	N135		0	10:20	29 N63	N22	N70	N134		0
10:25	30 N63	N26	N83	N135		0	10:25	30 N63	N22	N70	N134		0
10:30	31 N104	N27	N84	N138		0	10:30	31 N63	N22	N71	N134		0
10:35	32 N104	N27	N84	N138		0	10:35	32 N63	N24	N82	N131		0
10:40	33 N104	N27	N84	N138		0	10:40	33 N104	N24	N82	N131		0
10:45	34 N104	N28	N86	N139		0	10:45	34 N104	N24	N82	N131		0

Figure 11: An example of a schedule versus its output

5 STUDY RESULTS

This section discusses the findings of the study, which aimed to examine the impact of optimal scan scheduling on the energy consumption of the radiology department using maximum makespan and waiting time as key performance indicators. Section 5.1 evaluates the results using 25 replications to ensure accuracy, as declared in Section 4.3.7, and the effectiveness of the scheduling heuristics for CT1 and CTSEH by assessing the number of reserved scheduling blocks. Additionally, Section 5.2 shows the connection between energy consumption and the spared operating hours, which led to a reduction in wasteful idle state energy consumption.

5.1 RESERVED SCHEDULING BLOCKS AND MAKESPAN OUTCOME

By altering the number of reserved scheduling blocks per hour, the waiting time of urgent patients and the maximum completion time fluctuates. We have evaluated several schedules with different numbers of reserved blocks on CTSEH for both LEPT and SEPT but aiming to keep a similar service level for urgent patients. Figure 12 shows the base schedule with zero reserved blocks on CTSEH which results in a maximum waiting time of 3 hours and an average waiting time of 0.16 hours for urgent patients. The average maximum makespan of the LEPT heuristic is 40 hours and 33 minutes and that of the SEPT heuristic is 40 hours and 11 minutes, which nearly is a full workweek. The outcome of LEPT and SEPT is similar, which makes sense since there is little to no optimization possible with no second machine available for scheduling.

Reserved Timeslots						WT	Avg	Max	Min	CMAXLEPT	Avg	Max	Min	CMAXSEPT	Avg	Max	Min
	1	2	3	4	5		0.160007		3	0	40:33:36	48:30:00	31:30:00		40:11:00	47:40:00	31:30:00
8	0	0	0	0	0												
9	0	0	0	0	0												
10	0	0	0	0	0												
11	0	0	0	0	0												
12	0	0	0	0	0												
13	0	0	0	0	0												
14	0	0	0	0	0												
15	0	0	0	0	0												
16	0	0	0	0	0												

Figure 12: Base outcome with zero reserved scheduling blocks. The figure shows schedule 1 with its maximum, minimum, and average urgent patient waiting time and total makespan for LEPT and SEPT.

Figure 13 shows the simulation outcome with the calculated potentially optimal reserved scheduling blocks on CTSEH, creating room for optimization. The service level is impacted within limits. Urgent patients have an average waiting time of 0.57 hours and an increase of the maximum waiting time by 2 hours. The LEPT heuristic decreases the average makespan by 26% and the SEPT heuristic decreases the average makespan by 11%.

Reserved Timeslots						WT	Avg	Max	Min	CMAXLEPT	Avg	Max	Min	CMAXSEPT	Avg	Max	Min
	1	2	3	4	5		0.568624		5	0	30:06:24	34:50:00	23:50:00		35:31:48	41:05:00	27:30:00
8	2	2	2	2	1												
9	3	2	3	2	2												
10	2	2	2	1	2												
11	2	2	4	3	3												
12	7	7	8	7	7												
13	5	4	5	5	5												
14	6	5	5	5	5												
15	4	3	4	4	3												
16	2	1	1	1	1												

Figure 13: Simulation outcome with the calculated reserved scheduling blocks. The figure shows schedule 5 with its maximum, minimum, and average urgent patient waiting time and total makespan for LEPT and SEPT.

Appendix F shows 5 other schedules of reserved scheduling blocks. We have proposed those 5 other schedules on the basis of trial and error together with MST and have evaluated their outcomes to see how

these different reservation blocks influence the waiting time and total completion time. Table 8 and 9 show the simulation results for every schedule with the increase of the waiting time and decrease of makespan relative to schedule 1, the base schedule.

Table 8: Simulation outcome. Waiting Time (WT) in hours and C_{max} in hh:mm:ss.

Schedule	Average WT (h)	Max WT (h)	Average C_{max} LEPT (hh:mm:ss)	Average C_{max} SEPT (hh:mm:ss)
1	0.16	3	40:33:36	40:11:00
2	0.23	3	36:05:12	40:23:00
3	0.38	3	36:10:36	40:56:48
4	0.55	4	33:29:36	35:20:24
5	0.57	5	30:06:24	35:31:48
6	0.26	3	35:14:12	40:10:00
7	0.40	4	33:01:24	37:41:48

Table 9: Simulation outcome in relation to base schedule 1. All in percentage relative to base schedule 1.

Schedule	Average WT (%)	Max WT (%)	Average C_{max} LEPT (%)	Average C_{max} SEPT (%)
1	0.0%	0.0%	0.0%	0.0%
2	+43.8%	0.0%	-11.0%	+0.5%
3	+137.2%	0.0%	-10.8%	+1.9%
4	+242.9%	+33.3%	-17.4%	-12.1%
5	+255.4%	+66.7%	-25.8%	-11.6%
6	+60.6%	0.0%	-13.1%	0.0%
7	+147.2%	+33.3%	-18.6%	-6.2%

We see that in Table 8 the column 'Average C_{max} LEPT' consistently has a lower total completion time relative to schedule 1 and also consistently a lower total completion time than in column 'Average C_{max} SEPT'. The results of our study demonstrate that the LEPT heuristic consistently outperforms the SEPT heuristic in all schedules, consistent with the previous study of Arnaout (2010). While we observe a decrease in the average makespan, it is important to note that this is consistently accompanied by an increase in both the average and maximum waiting time. However, our findings suggest that the tradeoff between the reduction in machine utilization time and the increase in waiting time is favorable. Specifically, we observe that the average waiting time increases by less than one hour while saving seven hours on the machine. These results support the adoption of the proposed strategy and highlight the potential benefits of scheduling normal scans on CTSEH.

To confirm the statistical significance of our results, we conducted a two-tailed Student's T-test with $\alpha = 0.05$, comparing the average makespan obtained from the LEPT and SEPT heuristics to the base outcome from schedule 1, as well as to each other in relation to schedule 5. In both cases, we rejected the null hypothesis of no significant difference in favor of the alternative hypothesis. Appendix F.2 shows all T-tests.

5.2 ENERGY CONSUMPTION IMPACT

The current schedule results in at least 40 hours of scan time per workweek in our simulation. For a comparison with real-world data, the first 5-day workweek of 2019 resulted in at least 39.6 hours of scan time. The best-performing heuristic LEPT resulted in an average makespan of 30.1 hours of scan time per day, which saves 9.9 hours per week on average. This translates into 514.8 hours annually. To calculate how much energy we save, we have used data from Section 2.3 where is shown that the system idle state consumes around 32 kWh more per hour than the system off state. We have shown that using the LEPT heuristic, we can perform the same number of scans in less on-time on CT1, causing the operating hours to decrease and the system being able to be shut off earlier. This then results in less system idle state and less energy consumption. If we save 514.8 hours annually on CT1, this results in a decrease of 16473 kWh of energy consumption annually. This is 0.07% of MST's total energy consumption annually for only applying this heuristic to only one single machine. It also is equal to the average energy consumption of 5 three-person households per year in The Netherlands (Nibud, 2023).

6 CONCLUSIONS & RECOMMENDATIONS

The main goal of this research was to help MST gain insight into the CT scan scheduling influence on energy efficiency to become more sustainable and have an optimized schedule while also aiming to keep the same service level for patients. By bridging the knowledge gap between the accepted standard and reality, and also by creating a Mont Carlo simulation to test how to optimize the schedule, the aim is achieved to an extent within the radiology department. This section contains the conclusions of the research based on the research questions and also contains recommendations for MST to further continue its sustainability goal.

6.1 CONCLUSIONS

In conclusion, our research has shed light on the current CT scheduling practices at the radiology department of MST, which operate in a centralized and decentralized manner. Patients are scheduled through the administration office of the department or by doctors themselves and were not influenced by regulations. Data on patient arrivals, therefore, was difficult to collect. The stakeholders in the process were the lab technicians, physicists, administrative office employees, and radiologists. The weekly schedule was only able to be exported as a screenshot and patient scan duration could not be retrieved due to it not being logged. Existing data is retrieved through clinical physicists, business intelligence, building maintenance, and contracts & hospitality. There is no dedicated employee within the radiology department to retrieve data. Non-existing data such as the arrival rate per time period had to be measured or retrieved from similar literature to create accurate simulation input.

To accurately model the scheduling system, we selected the year 2019 as a representative period unaffected by COVID-19. From the collected data, we focused on the 20 most frequently occurring scan types and performed detailed calculations to determine their processing times. Empirical distributions were predominantly used, with a few exceptions following gamma and exponential distributions. Assumptions were made regarding scan durations, considering that a scan would conclude as soon as the next scan started, except during lunchtime or at the end of the day.

The scheduling problem at CT1 and CTSEH of the radiology department was formulated as $P_2|p_j|C_{max}$ where p_j represents the processing time of each job on the respective CT scanner. To optimize the scheduling process and minimize the total makespan, we employed two heuristics proposed by Arnaout (2010): Longest Expected Processing Time-first (LEPT) and Shortest Expected Processing Time-first (SEPT). Through a Monte Carlo simulation with 25 replications, we evaluated the performance of these heuristics. As anticipated, LEPT outperformed SEPT, achieving an average makespan of 30.1 hours compared to SEPT's 35.5 hours, using our calculated reserved scheduling blocks.

The implementation of the LEPT heuristic, with its improved scheduling efficiency, has the potential to significantly reduce energy consumption at CT1. Currently, CT1 consumes approximately 0.233 GWh per year, accounting for 1.03% of MST's total energy consumption. Interestingly, during a specific week, 38.2% of CT1's energy consumption was attributed to idle states. By implementing the recommended scheduling strategy, the average makespan can be reduced from approximately 40 hours to 30.1 hours per workweek, resulting in 9.9 fewer hours of idle state and an estimated annual energy consumption decrease of 16473 kWh. This corresponds to a 0.07% reduction in MST's overall energy consumption.

In summary, our research has demonstrated the potential of optimized scheduling strategies in improving energy efficiency and reducing CO₂ emissions in the radiology department at MST. By addressing the challenges in data collection and applying the LEPT heuristic, we have identified opportunities for significant energy savings while maintaining an acceptable level of waiting time. These findings contribute to the broader scientific understanding of scheduling optimization and its implications for sustainable healthcare practices. Our research offers valuable insights for MST and other healthcare institutions, supporting their sustainability objectives and paving the way for future studies in scheduling algorithms and energy consumption optimization.

6.2 RECOMMENDATIONS

This section contains recommendations for practical implementation and future research.

6.2.1 Practical implementation

This research shows the potential of both heuristics for the reduction of the total weekly completion time on CT1. The implementation of a schedule where the scans with the longest or shortest processing times are scheduled first during a whole week may seem impractical or nonsensical, but an alternate form can also be realized of the LEPT or SEPT heuristic. The scans with the largest processing times would then be set at the beginning of each day, where potential rescanning could cause more delay than similar rescanning with shorter scans. This delay is preferable earlier in the day since there is more time to reduce potential overtime, waiting time, or scan rescheduling than when the delay happens later during the day. This potentially results in a reduced reduction, but still holds the potential to optimize the current schedule. When such a heuristic is applied to the scan scheduling process, certain options in HiX to show the first available slot for a certain scan type could also be beneficial.

Also, we recommend that MST inform its radiology employees about the effects certain habits have regarding CT energy usage. Simply not shutting off the CT completely can make an impactful difference in energy consumption per day of the CT machine and the hospital, potentially also on the lifespan of the machine. Converting these effects into costs may yield higher comprehension of CT handling and the effects on wasteful energy consumption.

6.2.2 Future research

- Not a lot of practical data is available next to the number of scans performed per year. It can be of high importance for future decisions to log at the minimum the scan processing time, patient room arrival, patient urgency, and number of scan type requests per time period. Research using these accurate variables is then more practical. A dedicated employee within the administration office of the radiology department can be of high value.
- The heuristics are only applied in regard to CT1 and CTSEH and do not incorporate patient urgency. If planning on the use of urgency types, the heuristics can be changed to give priority to patients with a higher priority.
- The study does not incorporate the increase in useful energy consumption on CTSEH by performing more scans on the machine and only calculates the decrease of wasteful idle state energy consumption on CT1. For a total picture, this could be investigated further.
- The scope of this research is for two specific CT machines in the radiology department. If research is performed on the other CT machines within MST, it may bring forth more potential energy reduction.

- The research does not focus on explanations of why certain scans take longer than other scans and what potential factors cause scan delay. Researching and reducing these causes might also provide less unnecessary idle state and provide an energy reduction.
- Research into removing patient processes outside of the CT room can also cause a reduction of unnecessary system idle state and more actual scanning state and thus an increase in energy efficiency.

6.2.2.1 Future use

This research is based on several assumptions regarding input data. Once MST has more realistic data to enter into this simulation, the simulation could form a better basis to validate the scheduling heuristic and a better argument to change the current scheduling system to a new, improved system. The following elements are available to change with more realistic data:

- Scan processing time

This value is dependent on what distribution the respective scan processing time follows. When realistic processing times are available, the Sub ProcTimeGeneration found in Appendix D.1 should be disabled and every known processing time for a certain time period should be put in their respective worksheet in the workbook '2019 data analysis.xlsx' under the heading 'Duration' in row 3, column 48. Every scan type should get its own sheet. Then the Sub ScanGeneration in Appendix D.2 will retrieve accurate processing times per scan type for normal scans. Urgent scan processing times will have to be found using a similar code as the normal scans, but an individual with VBA knowledge will be able to implement it by looking at the code for normal scans.

- Number of scans per scan type & arrival rate

Currently, the number of scans is generated by taking the average number of scans per week in 2019 and applying arrival factors per hour and day to the average. Then the occurrence rate of the respective scan types is compared with the number of scans and a scan type is given to every generated scan. Ideally, the arrival rate per hour per scan type is known, so every scan type would then be generated individually, but this would require a different type of code.

To implement more accurate data into this model, the new cumulative occurrence rate per scan type should be added to worksheet 'CT 2019 CT1' and 'CT 2019 CTSEH' in workbook 'MC CT 19_20_21.xlsm' and sorted on occurrence rate. Then the total number of scan types should be altered in the Excel VBA procedure *ScanGeneration* from 133 and 107 in lines 41 and 43 to the new number of scan types. Also, a new worksheet should be created in workbook '2019 data analysis.xlsx' with the exact scan type SPS code and worksheet name, and processing time duration starting in row 3, column 48. The new average number of scans per week can be changed in lines 23, 31, and 34 from 133 and 230 to the new numbers.

The cumulative arrival rate factors are currently retrieved from research in a similar hospital from its echography department. These hourly, daily, and weekly factors are found in the sheet 'Monte Carlo Simulation' in the workbook 'MC CT 19_20_21.xlsm'. When more accurate values are found concerning the number of CT scan requests per day and hour, calculations should then be done to alter them to cumulative rates and then should be altered in their respective cell in the

worksheet. Weekly rates are not yet implemented in the code due to the research generating weekly patterns but can be by changing the original code of Sub *ScanGeneration* in the same manner as days and weeks are implemented. An extra total loop is necessary to loop over weeks as well.

If done correctly, the accurate cumulative weekly factors can be entered into the '*Monte Carlo Simulation*' worksheet. These can be relevant to simulate seasonality for example. The arrays *ScansN()* and *ScansU()* should be dimmed (1 to 6, ...) to add the weeks to the array and when printing the scans in the worksheet, an extra column will have to be added to both normal and urgent scans column clusters to print every week number. The procedure *RunSim* should also be adjusted to incorporate running over 365 days instead of 5 days and the procedure *ScheduleScans* should also incorporate the week numbers for appointment block searching.

It is recommended to let an individual with VBA knowledge apply the changes in the code.

BIBLIOGRAPHY

- Abdalkareem, Z. A., Amir, A., Azmi Al-Betar, M., Ekhan, P., & Hammouri, A. I. (2021). Healthcare scheduling in optimization context: a review. *Health and Technology*, 445-469.
- Arnaout, J.-P. (2010). Heuristics for the Maximization of Operating Rooms Utilization Using Simulation. *SIMULATION*, 573-583.
- Bawaneh, K., Ghazi Nezami, F., Rasheduzzaman, M., & Deken, B. (2019). Energy Consumption Analysis and Characterization of Healthcare Facilities in the United States. *Energies*, 3775.
- Birge, J. R., & Louveaux, F. (2011). *Introduction to Stochastic Programming*. New York: Springer.
- Borgman, N. J., Vliegen, I. M., Boucherie, R. J., & Hans, E. W. (2018). Appointment scheduling with unscheduled arrivals and reprioritization. *Flexible Services and Manufacturing Journal*, 30-53.
- Deceuninck, M., Fiems, D., & De Vuyst, S. (2018). Outpatient scheduling with unpunctual patients and no-shows. *European Journal of Operational Research*, 195-207.
- Field, A. (2009). *Discovering*. Sage publications.
- Frader, I., & Matthiesen, R. (2010). Overview on Techniques in Cluster Analysis. In R. Matthiesen, *Bioinformatics Methods in Clinical Research* (pp. 81-107). New Jersey: Humana Totowa.
- Friendly Power. (2021, February 1). *Business Energy Advisors | Hospital*. Retrieved from Biz Energy Advisor: <https://esource.bizenergyadvisor.com/article/hospitals>
- Fung, G. (2001). A comprehensive overview of basic clustering algorithms. 1-37.
- Graham, R. (1966). Bounds for certain multiprocessing anomalies. *Bell Labs Technical Journal*, 1563-1581.
- Graham, R. L., Lawler, E. L., Lenstra, J., & Rinnooy Kan, A. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. *Elsevier*, 5, 287-326.
- Gupta Strategists. (2019). *Een stuur voor de transitie naar duurzame gezondheidszorg*.
- Harrison, R. L. (2010). Introduction to monte carlo simulation. *AIP conference proceedings* (pp. 17-21). American Institute of Physics.
- Heale, R., & Twycross, A. (2015). Validity and reliability in quantitative studies. *Evid Based Nurs*, 66-67.
- Heerkens, H., & van Winden, A. (2017). *Solving Managerial Problems Systematically*. Noordhoff Uitgevers.
- Heye, T., Knoerl, R., Wehrle, T., Mangold, D., Cerminara, A., Loser, M., . . . Merkle, E. (2020). The Energy Consumption of Radiology: Energy- and Cost-saving Opportunities for CT and MRI Operation. *Radiology*, 295(3), 593-605.
- Hofman, L. (2014). *Capacity management at the radiology: Managing the variability of scheduled and unscheduled*. Enschede: University of Twente.
- Kamstra, K., & Schuppert, W. (2022, October 31). Schematic overview of power measurement. MST.

- Koeleman, P. M., & Koole, G. M. (2012). Optimal outpatient appointment scheduling with emergency arrivals and general service times. *IIE Transactions on Healthcare Systems Engineering*, 14-30.
- Lamiri, M., Xie, X., Dolgui, A., & Grimaud, F. (2008). A stochastic model for operating room planning. *European Journal of Operational Research*, 1026-1037.
- Lillis, A. M. (1999). A framework for the analysis of interview data from. *Accounting and Finance*, 79-105.
- Mahase, E. (2022). Hopeful not helpless: How doctors are combating climate change. *BMJ*.
- Mason, W. M., & Wolfinger, N. H. (2001). Cohort Analysis. *International Encyclopedia of Social and Behavioral Sciences*.
- McGain, F., & Naylor, C. (2014). Environmental sustainability in hospitals – a systematic review and research agenda. *Sage Journals*, 245-252.
- Medisch Spectrum Twente. (2022, September 9). *MST Jaarverslag 2021*. Retrieved from Medisch Spectrum Twente: <https://www.mst.nl/jaarverslag2021>
- Mohammad, A. E., Ashkan, J., Twomey, J., Yildirim, B., Overcash, M., Elsken, T., . . . Mcadam, A. (2011). Energy Consumption of VA Hospital CT Scans. *2011 IEEE International Symposium on Sustainable Systems and Technology* (pp. 1-5). IEEE.
- Mundform, D. J., Schaffer, J., Kim, M.-J., Shaw, D., Thongteeraparp, A., & Supawan, P. (2011). Number of Replications Required in Monte Carlo. *Journal of Modern Applied Statistical Methods*, 19-28.
- Namey, E., Guest, G., Thairu, L., & Johnson, L. (2008). Data Reduction Techniques for Large Qualitative Data Sets. In G. Guest, & K. M. MacQueen, *Handbook for team-based qualitative research* (pp. 137-161). Plymouth: AltaMira Press.
- Nibud. (2023, March 14). *Kosten van energie en water*. Retrieved from Nibud: https://www.nibud.nl/onderwerpen/uitgaven/kosten-energie-water/?gclid=Cj0KCQjwtsCgBhDEARIsAE7RYh3xe-Gbz13F57y1KEPVRk63RhndJy5_B8tmnZftPOPpAGIErS0y7UUaAmUZEALw_wcB
- Pinedo, M. (2012). *Scheduling: Theory, Algorithms, and Systems*. New York: Springer.
- Rijksoverheid. (2022, April 22). *Meer duurzaamheid in de zorg*. Retrieved from Rijksoverheid: <https://www.rijksoverheid.nl/onderwerpen/duurzame-zorg/meer-duurzaamheid-in-de-zorg>
- Ruiz, R. (2015). Scheduling Heuristics. In R. Martí, P. Panos, & M. Resende, *Handbook of Heuristics* (pp. 1-24). Springer, Cham.
- Sagnol, G., Waldschmidt, D. S., & Tesch, A. (2018). The Price of Fixed Assignments in Stochastic Extensible Bin Packing. *Lecture Notes in Computer Science* (pp. 327–347). Springer Science+Business Media.
- Shapiro, A. (2001). Monte Carlo simulation approach to stochastic programming. *Proceeding of the 2001 Winter Simulation Conference (Cat. No.01CH37304)*. Arlington: IEEE.
- Shapiro, A., Higle, J. L., Römisch, W., & Sen, S. (2000). *Stochastic programming by Monte Carlo simulation methods*. Berlin: Humboldt-Universität zu Berlin.

- Shen, C., Zhao, K., Ge, J., & Zhou, Q. (2018). Analysis of Building Energy Consumption in a Hospital. *10th International Conference on Applied Energy* (pp. 22-25). Hong Kong: Elsevier.
- Study.com. (2022, November 29). *Research Types & Examples | Exploratory, Descriptive, & Explanatory*. Retrieved from Study.com: <https://study.com/learn/lesson/research-types-examples-exploratory-descriptive-explanatory.html>
- Tufféry, S. (2011). *Data Mining and Statistics for Decision Making*. West Sussex: John Wiley & Sons, Ltd.
- Twente. (2022, April 21). Retrieved from Twente : <https://www.twente.com/organisaties/medisch-spectrum-twente>
- Types of Electrical Drawing and Diagrams*. (2022, October 31). Retrieved from Electrical Technology: <https://www.electricaltechnology.org/2020/04/types-electrical-drawing-diagrams.html>
- Vanhaecht, K., Panella, M., van Zelm, R., & Sermeus, W. (2010). An overview on the history and concept of care pathways as complex interventions. *International Journal of Care Pathways*, 117-123.
- Vergunst, F., Berry, H. L., Rugkåsa, J., Burns, T., Molodynski, A., & Maughan, D. L. (2020, February). Applying the triple bottom line of sustainability to healthcare research—a feasibility study. *International Journal for Quality in Health Care*, 32(1), 48-53.
- Vermeulen, I. B., Bohte, S. M., Elkhuisen, S. G., Lameris, J. S., J., B. P., & La Poutré, J. A. (2009). Adaptive Resource Allocation for Efficient Patient. *Artificial Intelligence in Medicine*, 67-80.

APPENDICES

APPENDIX A: INTERVIEW STRUCTURE

Appendix A.1: Interview Questions

Employees of the following departments have been interviewed:

- Radiology
- Administration (Radiology)
- Property Management
- Contracts Hospitality & Logistics
- Business Intelligence

The main department that has been interviewed, the department of Radiology, were asked the following questions:

- What are the different CT work shifts?
- How does such a shift look like in general?
- How has the current CT scan schedule been designed?
- What are the planned timeframes in the CT schedule?
- What factors influence a patient's scan time?
- How much time is asked to the patients in advance of a scan?
- What are the food and drink restrictions before a CT scan?
- Do you currently have ideas to potentially optimize the CT scan process?
- What processes within the CT scan room can potentially be done outside of the room?
- How much time is planned per scan type?
- How many lab technicians are there in total and how much per shift?
- How does the process work when an emergency patient enters the hospital?
- What data in regard to time is being logged?
- Is there knowledge or data about:
 - planned time versus actual time per scan?
 - involuntary waiting times?
 - no-shows?

APPENDIX B: CT SCHEDULE & CODES

Appendix B.1: CT Schedules

Appendix B.1.1: CT1 Schedule

ECT1 (RADIOLOGIE)						
	maandag 09 januari 2023	dinsdag 10 januari 2023	woensdag 11 januari 2023	donderdag 12 januari 2023	vrijdag 13 januari 2023	
0800	C1-02;C1-03;... Enschede MST	▲ C1-02;C1-03;... Enschede MST	C0-11;C5-59 Enschede MST	C1-12 Enschede MST	C1-02;C1-03;... Enschede MST	
10	▲ C1-02;C1-03;... Beschikbaar vanaf	C1-02;C1-03;... Enschede MST	C0-11;C5-59 Enschede MST	C1-12 Enschede MST	▲ C1-02;C1-03;... Beschikbaar vanaf	
20	C8-21 Enschede MST	▲ C1-02;C1-03;... Beschikbaar vanaf	C0-11;C5-59 Enschede MST	C1-12 Enschede MST	C8-21 Enschede MST	
30	C8-21 Enschede MST	C1-02;C1-03;... Enschede MST	C0-11;C5-59 Enschede MST	C1-12 Enschede MST	C8-21 Enschede MST	
40	C1-06;C5-30;... Enschede MST	C1-06;C5-30;... Enschede MST	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	C1-06;C5-30;... Enschede MST	
50	● Blokkade	● Blokkade	C0-11;C5-59 Enschede MST	● Blokkade		
0900	C1-02;C1-03;... Enschede MST	C0-01;C0-04;... Enschede MST	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	
10	C1-02;C1-03;... Enschede MST	C0-01;C0-04;... Enschede MST	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	▲ C1-02;C1-03;... Beschikbaar vanaf	
20	▲ C1-02;C1-03;... Beschikbaar vanaf	C0-01;C0-04;... Enschede MST	C0-11;C5-59 Enschede MST	▲ C1-02;C1-03;... Beschikbaar vanaf	C1-02;C1-03;... Enschede MST	
30	C1-02;C1-03;... Enschede MST	C0-01;C0-04;... Enschede MST	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	C1-06;C5-30;... Enschede MST	
40	C1-06;C5-30;... Enschede MST	● Blokkade	C0-11;C5-59 Enschede MST	C1-06;C5-30;... Enschede MST	● Blokkade	
50	● Blokkade	C1-06;C5-30;... Enschede MST	C0-11;C5-59 Enschede MST		▲ C1-02;C1-03;... Beschikbaar vanaf	
1000	▲ C1-02;C1-03;... Beschikbaar vanaf 5-1-2023 00:00:00	C0-01;C0-04;... Enschede MST	C1-06;C5-30;... Enschede MST	C1-06;C5-30;... Enschede MST	C1-02;C1-03;... Enschede MST	
10	C1-02;C1-03;... Enschede MST	C0-01;C0-04;... Enschede MST	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	
20	C1-02;C1-03;... Enschede MST	C0-01;C0-04;... Enschede MST	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	▲ C1-02;C1-03;... Beschikbaar vanaf	
30	▲ C1-02;C1-03;... Beschikbaar vanaf	▲ C1-02;C1-03;... Beschikbaar vanaf	C0-11;C5-59 Enschede MST	▲ C1-02;C1-03;... Beschikbaar vanaf	C1-02;C1-03;... Enschede MST	
40	C1-02;C1-03;... Enschede MST	● Blokkade	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST		
50			C0-11;C5-59 Enschede MST	● Blokkade	C5-53 Enschede MST	
1100	C5-53 Enschede MST	C5-53 Enschede MST	C0-11;C5-59 Enschede MST	C5-53 Enschede MST	C5-55 Enschede MST	
10	C5-55 Enschede MST	C5-55 Enschede MST	C0-11;C5-59 Enschede MST	C5-55 Enschede MST	C5-55 Enschede MST	
20	C5-57 Beschikbaar vanaf 8-1-2023	C1-02;C1-03;... Enschede MST	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	▲ C5-57 Beschikbaar vanaf 12-1-2023	
30	▲ C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	
40	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	▲ C5-57 Beschikbaar vanaf 10-1-2023	C1-02;C1-03;... Enschede MST	▲ C5-57 Beschikbaar vanaf 12-1-2023	
50	▲ C5-57 Beschikbaar vanaf 8-1-2023	C1-02;C1-03;... Enschede MST	● Blokkade	C1-02;C1-03;... Enschede MST	C1-06;C5-30;... Enschede MST Pre hydratatie	
1200	C1-06;C5-30;... Enschede MST Pre hydratatie	C1-02;C1-03;... Enschede MST Pre hydratatie	▲ C1-06;C5-30;... Beschikbaar vanaf 4-1-2023 00:00:00	C1-06;C5-30;... Enschede MST Pre hydratatie	C1-02;C1-03;... Enschede MST	
10	C1-02;C1-03;... Enschede MST	▲ C1-02;C1-03;... Beschikbaar vanaf 5-1-2023 00:00:00	C1-06;C5-30;... Enschede MST	C1-02;C1-03;... Enschede MST	C1-06;C5-30;... Enschede MST	
20	C1-06;C5-30;... Enschede MST	C1-06;C5-30;... Enschede MST	Pa ● Blokkade	C1-06;C5-30;... Enschede MST	C1-02;C1-03;... Enschede MST	
30	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST		C1-02;C1-03;... Enschede MST		
40	C1-02;C1-03;... Enschede MST					
50						
1300	C7-90;C7-91 Enschede MST	C0-11;C5-59 Enschede MST	C7-90;C7-91 Enschede MST	C7-90;C7-91 Enschede MST	C7-90;C7-91 Enschede MST	
10	C7-90;C7-91 Enschede MST	C0-11;C5-59 Enschede MST	C7-90;C7-91 Enschede MST	C7-90;C7-91 Enschede MST	C7-90;C7-91 Enschede MST	
20	C7-90;C7-91 Enschede MST	C0-11;C5-59 Enschede MST	C7-90;C7-91 Enschede MST	C7-90;C7-91 Enschede MST	C7-90;C7-91 Enschede MST	
30	▲ C1-02;C1-03;... Beschikbaar vanaf	C0-11;C5-59 Enschede MST	C1-06;C5-30;... Enschede MST	▲ C1-02;C1-03;... Beschikbaar vanaf	▲ C1-02;C1-03;... Beschikbaar vanaf	
40	C1-06;C5-30;... Enschede MST	C0-11;C5-59 Enschede MST	C1-06;C5-30;... Enschede MST	C1-06;C5-30;... Enschede MST	C1-06;C5-30;... Enschede MST	
50		C0-11;C5-59 Enschede MST				
1400	▲ C5-57 Beschikbaar vanaf 8-1-2023	C0-11;C5-59 Enschede MST	▲ C5-57 Beschikbaar vanaf 10-1-2023	C1-02;C1-03;... Enschede MST	▲ C5-57 Beschikbaar vanaf 12-1-2023	
10	C1-02;C1-03;... Enschede MST	C0-11;C5-59 Enschede MST	▲ C1-02;C1-03;... Beschikbaar vanaf	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	
20	▲ C1-02;C1-03;... Beschikbaar vanaf	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	▲ C1-02;C1-03;... Beschikbaar vanaf	▲ C1-02;C1-03;... Beschikbaar vanaf	
30	C1-02;C1-03;... Enschede MST	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	
40	C1-06;C5-30;... Enschede MST	C0-11;C5-59 Enschede MST	C1-06;C5-30;... Enschede MST	C1-06;C5-30;... Enschede MST	C1-06;C5-30;... Enschede MST	
50	● Blokkade	C0-11;C5-59 Enschede MST	● Blokkade	● Blokkade	● Blokkade	
1500	C1-06;C5-30;... Enschede MST	C1-06;C5-30;... Enschede MST	C1-06;C5-30;... Enschede MST	C1-06;C5-30;... Enschede MST	C1-06;C5-30;... Enschede MST	
10	C1-02;C1-03;... Enschede MST	C0-11;C5-59 Enschede MST	▲ C1-02;C1-03;... Beschikbaar vanaf	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	
20	C1-02;C1-03;... Enschede MST	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	
30	C1-02;C1-03;... Enschede MST	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	
40	▲ C1-06;C5-30;... Beschikbaar vanaf 4-1-2023 00:00:00	C0-11;C5-59 Enschede MST	▲ C1-06;C5-30;... Beschikbaar vanaf 6-1-2023 00:00:00	▲ C1-06;C5-30;... Beschikbaar vanaf 9-1-2023 00:00:00	▲ C1-06;C5-30;... Beschikbaar vanaf 10-1-2023 00:00:00	
50		C0-11;C5-59 Enschede MST				
1600	C1-02;C1-03;... Enschede MST	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	
10	C1-02;C1-03;... Enschede MST	C0-11;C5-59 Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	
20			● Blokkade			

Figure 14: HiX CT1 Schedule Screenshot

Appendix B.1.2: CTSEH Schedule

ECTSEH1 (RADIOLOGIE)					
	maandag 09 januari 2023	dinsdag 10 januari 2023	woensdag 11 januari 2023	donderdag 12 januari 2023	vrijdag 13 januari 2023
40					
45					
50					
55					
0800	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST
05					
10	Blokkade	Blokkade	Blokkade	Blokkade	Blokkade
15	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST	C1-02;C1-03;... Enschede MST
20					
25					
30	Blokkade	Blokkade	Blokkade	Blokkade	Blokkade
35					
40					
45	⚠ C1-02;C1-03;... Beschikbaar vanaf 3-1-2023 00:00:00	⚠ C1-02;C1-03;... Beschikbaar vanaf 6-1-2023 00:00:00	⚠ C1-02;C1-03;... Beschikbaar vanaf 5-1-2023 00:00:00	⚠ C1-02;C1-03;... Beschikbaar vanaf 6-1-2023 00:00:00	⚠ C1-02;C1-03;... Beschikbaar vanaf 9-1-2023 00:00:00
50					
55					
0900	Blokkade	Blokkade	Blokkade	Blokkade	Blokkade
05					
10					
15	⚠ C1-02;C1-03;... Beschikbaar vanaf 4-1-2023 00:00:00	⚠ C1-02;C1-03;... Beschikbaar vanaf 6-1-2023 00:00:00	⚠ C1-02;C1-03;... Beschikbaar vanaf 6-1-2023 00:00:00	⚠ C1-02;C1-03;... Beschikbaar vanaf 9-1-2023 00:00:00	⚠ C1-02;C1-03;... Beschikbaar vanaf 10-1-2023 00:00:00
20					
25					
30	Blokkade	Blokkade	Blokkade	Blokkade	Blokkade
35					
40					
45	⚠ C1-02;C1-03;... Beschikbaar vanaf 5-1-2023 00:00:00	⚠ C1-02;C1-03;... Beschikbaar vanaf 5-1-2023 00:00:00	⚠ C1-02;C1-03;... Beschikbaar vanaf 9-1-2023 00:00:00	⚠ C1-02;C1-03;... Beschikbaar vanaf 10-1-2023 00:00:00	⚠ C1-02;C1-03;... Beschikbaar vanaf 11-1-2023 00:00:00
50					
55					
1000					
05					
10					

Figure 15: HiX CTSEH Schedule Screenshot

Appendix B.1.3: CT Blanco Schedule

[illegible]

Figure 16: HiX CT Blanco Schedule Screenshot

Appendix B.2 : Scan Type Codes

Appendix B.2.1: Scan type codes (C-Codes)

Agenda	Subagenda	Code	Duur	C	Contactwijze	Direct	Indirect	Globaal	Omschrijving	Kleur	Vervallen	Standaard verrichtingen	Directe verrichtingen	Indirecte verrichtingen
RADIOLOGIE		96-99R	15 V			0	0	—	Arthrografie enkel re		—			
RADIOLOGIE		C0-01	40 V			0	0	—	CT punctie abdomen		—			
RADIOLOGIE		C0-04	40 V			0	0	—	CT abcdesdrainage		—			
RADIOLOGIE		C0-10	40 V			0	0	—	CT punctie thorax		—			
RADIOLOGIE		C0-11	5 V			0	0	—	CT beoordeling cardioloog		—			
RADIOLOGIE		C0-12	5 V			0	0	—	CT COVID		—			
RADIOLOGIE		C0-20	40 V			0	0	—	CT punctie extremititeit		—			
RADIOLOGIE		C1-02	10 V			0	0	—	CT cerebrum + contrast		—			
RADIOLOGIE		C1-03	10 V			0	0	—	CTV cerebrum		—			
RADIOLOGIE		C1-04	5 V			0	0	—	CTA hersenen TIA poli		—			
RADIOLOGIE		C1-05	10 V			0	0	—	CT trombolysse		—			
RADIOLOGIE		C1-06	15 V			0	0	—	CT trombolysse +		—			
RADIOLOGIE		C1-07	5 V			0	0	—	CT cerebrum geheugenpoli		—			
RADIOLOGIE		C1-08	10 V			0	0	—	CT brughoek + contrast		—			
RADIOLOGIE		C1-09	10 V			0	0	—	CTA cerebrum		—			
RADIOLOGIE		C1-11	5 V			0	0	—	CT cerebrum blanco		—			
RADIOLOGIE		C1-12	15 V			0	0	—	CT stereotaxie		—			
RADIOLOGIE		C2-02	5 V			0	0	—	CT sinus		—			
RADIOLOGIE		C2-03	5 V			0	0	—	CT aangezicht		—			
RADIOLOGIE		C2-04	10 V			0	0	—	CT hals		—			
RADIOLOGIE		C2-05	5 V			0	0	—	CT petrosium/mastoid		—			
RADIOLOGIE		C2-09	10 V			0	0	—	CTA carotiden		—			
RADIOLOGIE		C2-10	10 V			0	0	—	CT orbita		—			
RADIOLOGIE		C3-01	5 V			0	0	—	CT LWK		—			
RADIOLOGIE		C3-10	5 V			0	0	—	CT CWK		—			
RADIOLOGIE		C3-20	5 V			0	0	—	CT THWK		—			
RADIOLOGIE		C4-01L	5 V			0	0	—	CT schouder li		—			
RADIOLOGIE		C4-01R	5 V			0	0	—	CT schouder re		—			
RADIOLOGIE		C4-02	5 V			0	0	—	CT skeletstatus		—			
RADIOLOGIE		C4-04	5 V			0	0	—	CT spieren		—			
RADIOLOGIE		C4-20L	5 V			0	0	—	CT elleboog li		—			
RADIOLOGIE		C4-20R	5 V			0	0	—	CT elleboog re		—			
RADIOLOGIE		C4-40B	5 V			0	0	—	CT pols bdz		—			
RADIOLOGIE		C4-40L	5 V			0	0	—	CT pols li		—			
RADIOLOGIE		C4-40R	5 V			0	0	—	CT pols re		—			
RADIOLOGIE		C5-01	10 V			0	0	—	CT thorax HR		—			
RADIOLOGIE		C5-02	10 V			0	0	—	CT thorax + contrast		—			
RADIOLOGIE		C5-03	5 V			0	0	—	CT thorax blanco		—			
RADIOLOGIE		C5-04	5 V			0	0	—	CT thorax HR blanco		—			
RADIOLOGIE		C5-30	15 V			0	0	—	CTA pulmonalis		—			
RADIOLOGIE		C5-40	15 V			0	0	—	CT calciumscore		—			
RADIOLOGIE		C5-51	10 V			0	0	—	CT aorta totaal		—			
RADIOLOGIE		C5-52	10 V			0	0	—	CT aorta thoracaal		—			
RADIOLOGIE		C5-53	15 V			0	0	—	CT TAVI		—			
RADIOLOGIE		C5-54	15 V			0	0	—	CT hartklep		—			
RADIOLOGIE		C5-55	15 V			0	0	—	CT longvenen		—			
RADIOLOGIE		C5-57	10 V			0	0	—	CT thorax + bovenbuik		—			
RADIOLOGIE		C5-59	20 V			0	0	—	CTA coronairen		—			
RADIOLOGIE		C5-80	10 V			0	0	—	CT thorax + pancreas		—			
RADIOLOGIE		C5-81	15 V			0	0	—	CT thorax + nieren		—			
RADIOLOGIE		C6-03	5 V			0	0	—	CT sc-gewricht		—			
RADIOLOGIE		C6-04B	45 V			0	0	—	CT mammo draadlokalisatie bdz		—			
RADIOLOGIE		C6-04L	45 V			0	0	—	CT mammo draadlokalisatie li		—			
RADIOLOGIE		C6-04R	45 V			0	0	—	CT mammo draadlokalisatie re		—			
RADIOLOGIE		C7-01	10 V			0	0	—	CT abdomen + contrast		—			
RADIOLOGIE		C7-04	10 V			0	0	—	CT hals + thorax + abdomen		—			
RADIOLOGIE		C7-05	5 V			0	0	—	CT abdomen blanco		—			
RADIOLOGIE		C7-08	10 V			0	0	—	CT thorax + abdomen + contrast		—			
RADIOLOGIE		C7-16	10 V			0	0	—	CTA buikvaten		—			
RADIOLOGIE		C7-18	15 V			0	0	—	CT nieren		—			
RADIOLOGIE		C7-20	15 V			0	0	—	CT lever (4 fasen)		—			
RADIOLOGIE		C7-30	10 V			0	0	—	CT aorta abdominaal		—			
RADIOLOGIE		C7-31	10 V			0	0	—	CT aorta DIEP		—			
RADIOLOGIE		C7-36	10 V			0	0	—	CTA aorta + bekken + benen		—			
RADIOLOGIE		C7-70	10 V			0	0	—	CT pancreas		—			
RADIOLOGIE		C7-90	30 V			0	0	—	CT colografie		—			
RADIOLOGIE		C7-91	30 V			0	0	—	CT colografie + contrast		—			
RADIOLOGIE		C8-21	20 V			0	0	—	CT bijnier + contrast		—			
RADIOLOGIE		C8-22	5 V			0	0	—	CT bijnier blanco		—			
RADIOLOGIE		C9-01	5 V			0	0	—	CT bekken		—			
RADIOLOGIE		C9-03	5 V			0	0	—	CT rotatieprotocol		—			
RADIOLOGIE		C9-10L	5 V			0	0	—	CT bovenbeen li		—			
RADIOLOGIE		C9-10R	5 V			0	0	—	CT bovenbeen re		—			
RADIOLOGIE		C9-20L	5 V			0	0	—	CT knie li		—			
RADIOLOGIE		C9-20R	5 V			0	0	—	CT knie re		—			
RADIOLOGIE		C9-30L	5 V			0	0	—	CT onderbeen li		—			
RADIOLOGIE		C9-30R	5 V			0	0	—	CT onderbeen re		—			
RADIOLOGIE		C9-96L	5 V			0	0	—	CT enkel/voet li		—			
RADIOLOGIE		C9-96R	5 V			0	0	—	CT enkel/voet re		—			
RADIOLOGIE		CT-01	30 V			0	0	—	CT whole body		—			

Figure 17 : HiX Screenshot Scan Type Codes (C-Codes)

RADIOLOGIE	CT blanco	C0-11/C0-12/C1-04/C1-11/C2-02/C2-03/C2-05/C1-03/C1-10/C3-20/C4-01U/C4-01R/C4-02/C4-20U/C4-20R/C4-40B/C4-40/C4-40R/C5-03/C6-03/C7-05/C8-22/C9-01/C9-03/C9-10U/C9-10R/C9-20U/C9-20R/C9-30U/C9-30R/C9-96U/C4-04/C7-09/C5-04
RADIOLOGIE	CT contrast 10 min	C1-02/C1-03/C1-05/C1-09/C2-04/C2-09/C1-10/C5-01/C5-02/C5-51/C5-52/C5-57/C5-80/C7-01/C7-04/C7-08/C7-16/C7-30/C7-31/C7-36/C7-70/00-04/C1-08
RADIOLOGIE	CT contrast 15 min	C1-06/C5-30/C5-40/C5-53/C5-54/C5-81/C7-06/C7-18/C7-20/00-04/00-05
RADIOLOGIE	CT SEH contrast	C1-02/C1-03/C1-09/C2-04/C2-09/C2-10/C5-02/C5-30/C5-52/C5-57/C5-80/C7-01/C7-04/C7-08/C7-16/C7-30/C7-31/C7-36/C7-70/00-04



APPENDIX D: MONTE CARLO SIMULATION

Appendix D.1: Processing Time VBA Code

```
Sub ProcTimeGeneration()  
  
Dim I, j As Integer  
Dim NrOfScans As Integer  
Dim NrOfBins As Integer  
Dim ScanType As String  
Dim Random As Double  
Dim Duration As Double  
Dim BinLB As Double  
Dim BinUB As Double  
Dim CumProbLB As Double  
Dim CumProbUB As Double  
Dim wks As Excel.Worksheet  
Dim wkb As Excel.Workbook  
  
Application.ScreenUpdating = False  
Application.Calculation = xlManual  
  
Set wkb = Excel.Workbooks("2019 data analysis.xls")  
''Generate processing times of scan types by searching their distribution type in worksheet name  
For Each wks In wkb.Worksheets  
    With wks  
  
If .Name Like "*"Gamma" Then  
  
        ''Average number as CT scans per scan type per year from 2019-2021  
        NrOfScans = wks.Cells(Rows.Count, 1).End(xlUp).Row-- 1  
  
        ''For loop over number of scans  
        For i = 1 To NrOfScans  
            ''Duration setup, random number to assign bin, set bin amount  
            Duration = 0  
            Random = Rnd()  
            ''For loop to set bin and 57column57nce57y bounds  
            For j = 1 To NrOfBins  
                ''Give value if constraints are met  
                Duration = WorksheetFunction.Gamma_Inv(Random, wks.Cells(19, 24), wks.Cells(20, 24))  
                ''Value is found, so exit loop  
                Exit For  
            Next j  
            ''Print value to worksheet  
            wks.Cells(2 + i, 48) = Duration  
        Next i  
  
ElseIf .Name Like "*"Exp" Then  
  
        ''Average number as CT scans per scan type per year from 2019-2021  
        NrOfScans = wks.Cells(Rows.Count, 1).End(xlUp).Row-- 1  
  
        ''For loop over number of scans  
        For i = 1 To NrOfScans  
            ''Duration setup, random number to assign bin, set bin amount  
            Duration = 0  
            Random = Rnd()  
            NrOfBins = wks.Cells(Rows.Count, 28).End(xlUp).Row-- 2  
            ''For loop to set bin and 57column57nce57y bounds  
            For j = 1 To NrOfBins  
                ''Give value if constraints are met  
                Duration = (-1 * WorksheetFunction.Ln((1-- Random)) / (1 / wks.Cells(4, 24)))  
                ''Value is found, so exit loop  
                Exit For  
            Next j  
            ''Print value to worksheet  
            wks.Cells(2 + i, 48) = Duration  
        Next i  
  
ElseIf .Name Like "*"~" Then  
  
        ''Average number as CT scans per scan type per year from 2019-2021  
        NrOfScans = wks.Cells(Rows.Count, 1).End(xlUp).Row-- 1  
  
        ''For loop over number of scans  
        For i = 1 To NrOfScans  
            ''Duration setup, random number to assign bin, set bin amount  
            Duration = 0  
            Random = Rnd()  
        Next i  
    End With  
End For  
End Sub
```

```

NrOfBins = wks.Cells(Rows.Count, 43).End(xlUp).Row-- 2
''For loop to set bin and 58 column 58 nce 58 y bounds
For j = 1 To NrOfBins
    BinLB = wks.Cells(2 + j, 43)
    BinUB = wks.Cells(2 + j, 44)
    ''First bin has no left CumProb, so is set to 0
    If j = 1 Then
        CumProbLB = 0
        CumProbUB = wks.Cells(2 + j, 46)
        GoTo Calculate
    Else
        CumProbLB = wks.Cells(1 + j, 46)
        CumProbUB = wks.Cells(2 + j, 46)
    End If
End If

Calculate:
If Random > CumProbLB And Random <= CumProbUB Then
    ''Give value if constraints are met
    Duration = WorksheetFunction.RandBetween(BinLB, BinUB)
    ''Value is found, so exit loop
    Exit For
End If

Next j
''Print value to worksheet
wks.Cells(2 + i, 48) = Duration
Next i

Else
End If
End With
Next wks

Application.ScreenUpdating = True
Application.Calculation = xlAutomatic

End Sub

```

Appendix D.2: Scan Generation VBA Code

```

Sub ScanGeneration(ByVal It As Integer)

Dim i, j, lastRow, day, hour As Integer
Dim ScanTypeN, ScanTypeU As String
Dim Random As Double
Dim ScanTypeProcTime As Double
Dim ScanTypes(), z, LambdaN, LambdaU, ScanOccuranceN, ScanOccuranceU As Variant
Dim wks As Excel.Worksheet
Dim wkb As Excel.Workbook
Dim k, RandomScanTypeCell, ScanAmount, lastCol As Integer
''Fill the array ScanTypes and clear the generated scans from the last simulation
ScanTypes() = ThisWorkbook.Worksheets("CT 2019 CT").Range("V3:V2").Value
lastRow = Application.WorksheetFunction.Max(ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(Rows.Count, 11).End(xlUp).Row, 4)
ThisWorkbook.Worksheets("Monte Carlo Simulatio").Range("G4:" & lastRow).ClearContents
ThisWorkbook.Worksheets("Monte Carlo Simulatio").Range("G4:" & lastRow).Borders.LineStyle = xlNone
lastRow = Application.WorksheetFunction.Max(ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(Rows.Count, 17).End(xlUp).Row, 4)
ThisWorkbook.Worksheets("Monte Carlo Simulatio").Range("M4:" & lastRow).ClearContents
ThisWorkbook.Worksheets("Monte Carlo Simulatio").Range("M4:" & lastRow).Borders.LineStyle = xlNone
''Define multidimensional Lambda and ScanOccurance array with input parameters
ReDim LambdaN(1 To 5, 8 To 16), LambdaU(1 To 5, 8 To 16) As Variant
ReDim ScanOccuranceN(1 To 133, 1 To 5), ScanOccuranceU(1 To 107, 1 To 5) As Variant
ReDim ScansN(1 To 5, 1 To 2), ScansU(1 To 5, 1 To 2) As Variant
''Fill the Lambda Array with their respective lambd's per day and hour and fill 58 column 58 nce array
For day = 1 To 5
    For hour = 8 To 16
        ''134 = average number of scans per working week during operating hours in 2019 on CT1
        ''230 = average number of scans per working week during operating hours in 2019 on CTSEH
        LambdaN(day, hour) = (136 * _
            ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(14 + hour, 3) * _
            ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(33 + day, 3)) _
        LambdaU(day, hour) = (230 * _
            ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(14 + hour, 4) * _
            ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(33 + day, 4))
    Next hour
Next day
''Fill the scan 58 column 58 nce arrays with data
For j = 1 To 133
    For i = 1 To 5
        If j <= 107 Then
            ScanOccuranceU(j, i) = ThisWorkbook.Worksheets("CT 2019 CTSE").Cells(2 + j, i + 24)

```

```

        End If
        ScanOccuranceN(j, i) = ThisWorkbook.Worksheets("CT 2019 CT").Cells(2 + j, i + 24)
    Next i
Next j
''Generate scans per hour and day for a general working week for normal patients
For day = 1 To 5
    For hour = 8 To 16
        ''Generate random variable to choose a value from the Poisson arrival distribution
        Random = Rnd()
        For i = 1 To Application.Run("'xRealStats.xls'!Poisson_IN", Random, LambdaN(day, hour))
            ''Setup
            DurationN = 0
            ''Generate random variable to assign scan type to scan
            Random = Rnd()
            ''Find out what Scan Type the generated scan is through probability
            For j = 1 To 133

                If (Random > ScanOccuranceN(j, 3) And Random <= ScanOccuranceN(j, 4)) Then
                    ScanTypeN = ScanOccuranceN(j, 1)
                    Set wkb = Excel.Workbooks("2019 data analysis.xls")
                    ''Loop through all 20 chosen Scan Types
                    For k = 1 To UBound(ScanTypes)
                        z = Application.Match(ScanTypeN, ScanTypes, 0)
                        If Not IsError(z) Then
                            For Each wks In wkb.Worksheets
                                ''Assign scan duration to scantype
                                If InStr(wks.Name, ScanTypeN) > 0 Then
                                    ScanAmount = wks.Cells(Rows.Count, 48).End(xlUp).Row
                                    RandomScanTypeCell = WorksheetFunction.RandBetween(3,
ScanAmount)

                                    DurationN = wks.Cells(RandomScanTypeCell, 48)
                                    Exit For
                                End If
                            Next
                        ''If not one of the 20 chosen, evaluated scan types,
                        '' assign general given duration of scan type
                        Else
                            DurationN = ScanOccuranceN(j, 5)
                        End If
                    Exit For
                Next k
            Exit For
        End If
    Next j

    ''Fill the scan array with scans to schedule later
    ReDim Preserve ScansN(1 To 5, 1 To (UBound(ScansN, 2) + 1)) As Variant
    lastCol = LastNonZeroColumn(ScansN)
    ScansN(1, lastCol + 1) = "" & lastCol + 1
    ScansN(2, lastCol + 1) = ScanTypeN
    ScansN(3, lastCol + 1) = DurationN
    ScansN(4, lastCol + 1) = day
    ScansN(5, lastCol + 1) = hour

    ''Only print in worksheet when the last replication is performed
    If It = ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(7, 4) Then
        ''Print scannumber, type, processing time, day and hour
        lastRow = ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(Rows.Count, 7).End(xlUp).Row
        If ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow, 7) = "Sca" Then
            ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow + 1, 7) = "N"
        Else
            ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow + 1, 7) = "" & (lastRow-- 2)
        End If
        ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow + 1, 8) = ScanTypeN
        ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow + 1, 9) = DurationN
        ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow + 1, 10) = day
        ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow + 1, 11) = hour
    End If

Next i

''Generate scans per hour and day for a general working week for urgent patients
''Generate random variable to choose a value from the Poisson arrival distribution
Random = Rnd()
For i = 1 To Application.Run("'xRealStats.xls'!Poisson_IN", Random, LambdaU(day, hour))
    ''Setup
    DurationU = 0
    ''Generate random variable to assign scan type to scan
    Random = Rnd()
    ''Find out what Scan Type the generated scan is through probability
    For j = 1 To 107

```

```

        If (Random > ScanOccuranceU(j, 3) And Random <= ScanOccuranceU(j, 4)) Then
            ScanTypeU = ScanOccuranceU(j, 1)
            DurationU = ScanOccuranceU(j, 5)
        Exit For
    End If
Next j

''Fill the scan array with scans to schedule later
ReDim Preserve ScansU(1 To 5, 1 To (uBound(ScansU, 2) + 1)) As Variant
lastCol = LastNonZeroColumn(ScansU)
ScansU(1, lastCol + 1) = "" & lastCol + 1
ScansU(2, lastCol + 1) = ScanTypeU
ScansU(3, lastCol + 1) = DurationU
ScansU(4, lastCol + 1) = day
ScansU(5, lastCol + 1) = hour

''Only print in worksheet when the last replication is performed
If It = ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(7, 4) Then
    ''Print scannumber, type, processing time, day and hour
    lastRow = ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(Rows.Count, 13).End(xlUp).Row
    If ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow, 13) = "Sca" Then
        ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow + 1, 13) = ""
    Else
        ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow + 1, 13) = "" & (lastRow - 2)
    End If
    ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow + 1, 14) = ScanTypeU
    ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow + 1, 15) = DurationU
    ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow + 1, 16) = day
    ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(lastRow + 1, 17) = hour
End If

Next i
Next hour
Next day
''Add border to data
If It = ThisWorkbook.Worksheets("Monte Carlo Simulatio").Cells(7, 4) Then
    lastRow = Application.WorksheetFunction.Max(ThisWorkbook.Worksheets("Monte Carlo
Simulatio").Cells(Rows.Count, 11).End(xlUp).Row, 4)
    ThisWorkbook.Worksheets("Monte Carlo Simulatio").Range("G4:" & lastRow).BorderAround
LineStyle:=xlContinuous, Weight:=xlThin
    lastRow = Application.WorksheetFunction.Max(ThisWorkbook.Worksheets("Monte Carlo
Simulatio").Cells(Rows.Count, 17).End(xlUp).Row, 4)
    ThisWorkbook.Worksheets("Monte Carlo Simulatio").Range("M4:" & lastRow).BorderAround
LineStyle:=xlContinuous, Weight:=xlThin
End If

End Sub

```

Appendix D.3: Scan Scheduling + Calculation Procedures VBA Code

```

Sub ScheduleScans(ByVal It As Integer)

Const numDays As Integer = 5
Const numHours As Variant = 8 + (2 / 6)
Const numMinutesPerBlock As Integer = 5
Const numBlocksPerHour As Integer = 12
Dim wks As Excel.Worksheet
Dim firstIndex As Long
Dim numReplications, lastRow, consecutiveBlocks, necessaryblocks, actualblocks, lastnonzerovar As Integer
Dim ScheduleCTSEH(1 To numDays, 8 To 19, 1 To numBlocksPerHour), lastNonZeroIndices, matchIndex, allScans() As
Variant
Dim ScheduleCT1LEPT(1 To numDays, 8 To 18, 1 To numBlocksPerHour) As Variant
Dim ScheduleRealizedCT1LEPT(1 To numDays, 8 To 24, 1 To numBlocksPerHour) As Variant
Dim ScheduleRealizedCT1SEPT(1 To numDays, 8 To 24, 1 To numBlocksPerHour) As Variant
Dim ScheduleCTSEHLEPT(1 To numDays, 8 To 19, 1 To numBlocksPerHour) As Variant
Dim ScheduleRealizedCTSEHLEPT(1 To numDays, 8 To 24, 1 To numBlocksPerHour) As Variant
Dim ScheduleRealizedCTSEHSEPT(1 To numDays, 8 To 24, 1 To numBlocksPerHour) As Variant
Dim ScheduleCT1SEPT(1 To numDays, 8 To 18, 1 To numBlocksPerHour) As Variant
Dim ScheduleCTSEHSEPT(1 To numDays, 8 To 19, 1 To numBlocksPerHour) As Variant
Dim i, j, k, startk, startm, m, oldm, n, o, p, q, r, s, block, arrivalDay, arrivalHour, blockStart, blockEnd,
duration, scheduledduration, blockStart2 As Integer
Dim scanNumber As String
Dim foundZero As Boolean
Dim Cmax, Cmax2, Cmax3, Cmax4, Cmax5, Cmax6, Cmax7, Cmax8 As Date

Set wks = ThisWorkbook.Worksheets("Monte Carlo Simulation")

''Array to keep track of reserved blocks for CT1 scheduling
Dim ReservedCTSEH(1 To numDays, 8 To 19, 1 To numBlocksPerHour) As Integer

```

```

'Set the reserved scheduling blocks in replication 1
If It = 1 Then
    For j = 1 To numDays
        For k = 8 To 16
            If wks.Cells(152 + k, 36 + j) = 0 Then Exit For
            If k = 16 Then
                For m = 4 To (4 - wks.Cells(152 + k, 36 + j) + 1) Step -1
                    ReservedCTSEHBASIC(j, k, m) = 1
                Next m
            Else
                For m = 12 To (12 - wks.Cells(152 + k, 36 + j) + 1) Step -1
                    ReservedCTSEHBASIC(j, k, m) = 1
                Next m
            End If
        Next k
    Next j
End If

'Clear contents
wks.Range("U13:Y144").ClearContents
wks.Range("U148:Y291").ClearContents
wks.Range("AC13:AG156").ClearContents
wks.Range("AK13:AO156").ClearContents
wks.Range("AU13:AY144").ClearContents
wks.Range("AU148:AY291").ClearContents

For j = 0 To 7
    Worksheets("Schedule vs. Output").Range(Worksheets("Schedule vs. Output").Cells(6, 4 + 8 * j),
    Worksheets("Schedule vs. Output").Cells(149, 8 + 8 * j)).ClearContents
Next j

'Reset the schedule for each replication
For j = 1 To numDays
    For k = 8 To 24
        For m = 1 To numBlocksPerHour
            If k <= 18 Then
                ScheduleCT1LEPT(j, k, m) = 0
                ScheduleCT1SEPT(j, k, m) = 0
            End If

            If k <= 19 Then
                ScheduleCTSEH(j, k, m) = 0
                ScheduleCTSEHLEPT(j, k, m) = 0
                ScheduleCTSEHSEPT(j, k, m) = 0
            End If

            If k <= 16 Then
                ReservedCTSEH(j, k, m) = ReservedCTSEHBASIC(j, k, m)
            End If

            ScheduleRealizedCT1LEPT(j, k, m) = 0
            ScheduleRealizedCT1SEPT(j, k, m) = 0
            ScheduleRealizedCTSEHLEPT(j, k, m) = 0
            ScheduleRealizedCTSEHSEPT(j, k, m) = 0
        Next m
    Next k
Next j

'CT1
'LEPT

' Sort the array based on expected scantime and system arrival time
Call Sort2DArrayDesc(ScansN)

'Iterate over the generated scans
For j = LBound(ScansN, 2) To LastNonZeroColumn(ScansN)
    'Get the scan information
    scanNumber = ScansN(1, j)
    duration = ScansN(3, j)
    arrivalDay = ScansN(4, j)
    arrivalHour = ScansN(5, j)
    scheduledduration = ScansN(6, j)
    necessaryblocks = Application.WorksheetFunction.RoundUp((scheduledduration / (numMinutesPerBlock * 60)), 0)
    actualblocks = Application.WorksheetFunction.RoundUp((duration / (numMinutesPerBlock * 60)), 0)

    'Find the next available block in the schedule and reset the variables
    consecutiveBlocks = 0
    q = 0
    r = 0
    startk = 0

```

```

startm = 0

For o = 1 To 5
    For k = 8 To 16
        For m = 1 To numBlocksPerHour
            ' If k is 16 and m is greater than 4, exit the m loop, since no scheduling is done after 16:20
            If (k = 16 And m > 4) Then Exit For
            'Search only if the schedule is empty, not reserved and a scan does not take longer than 3
            If (ReservedCTSEH(o, k, m) = 1 And ScheduleCTSEHLEPT(o, k, m) = 0 And necessaryblocks <= 3)
                Then
                    For n = consecutiveBlocks To necessaryblocks - 1
                        ' If k is 16 and m + n - q is greater than 4, exit the m loop, since no scheduling
                        is done after 16:20
                        If (k = 16 And m + n - q > 4) Then Exit For
                        ' Check if current block is last block of the current hour and if the consecutive
                        blocks are less than necessaryblocks
                        If m + n - q > numBlocksPerHour And consecutiveBlocks < necessaryblocks Then
                            q = consecutiveBlocks
                            r = 1
                            oldm = m
                            startk = k
                            startm = m
                            'Continue to next hour and remember the start hour and block for the scheduling
                            Exit For
                        End If
                        'If the block is reserved and the schedule is empty, remember how many blocks are
                        found and look for the next block
                        If ReservedCTSEH(o, k, m + n - q) = 1 And ScheduleCTSEHLEPT(o, k, m + n - q) = 0
                            Then
                                consecutiveBlocks = consecutiveBlocks + 1
                            Else
                                consecutiveBlocks = 0
                                Exit For
                            End If
                        Next n
                        ' If k is 16 and m + n - q is greater than 4, exit the m loop, since no scheduling
                        is done after 16:20
                        If (k = 16 And m + n - q > 4) Then Exit For
                        'When arrived at the last block of the hour, go to the next hour
                        If m + n > numBlocksPerHour And consecutiveBlocks < necessaryblocks Then Exit For

                        'If the necessary blocks equal the found blocks, then schedule the scan using the first
                        hour a scan can be scheduled in if a scan overlaps multiple hours.
                        'Otherwise the code would use the first block from the next hour and not schedule
                        correctly.
                        If consecutiveBlocks = necessaryblocks Then
                            If startk = 0 Then
                                blockStart = ((k - 8) * numBlocksPerHour) + m
                            Else
                                blockStart = ((startk - 8) * numBlocksPerHour) + startm
                            End If
                            'Schedule the scan in the chosen blocks
                            For p = blockStart To blockStart + necessaryblocks - 1
                                ScheduleCTSEHLEPT(o, 8 + ((p - 1) \ numBlocksPerHour), (p - 1) Mod
numBlocksPerHour + 1) = scanNumber
                                Next p
                                'Go to the next scan
                                GoTo NextStep
                            End If
                        Else
                            'If value not found on the CTSEH machine, then reset values and continue the search in
                            the schedule for CT1
                            If r = 1 Then
                                q = 0
                                consecutiveBlocks = 0
                                r = 0
                                k = k - 1
                                m = oldm
                                startk = 0
                                startm = 0
                            End If
                            If ScheduleCT1LEPT(o, k, m) = 0 Then
                                'Look in ScheduleCT1
                                For n = consecutiveBlocks To necessaryblocks - 1
                                    ' If k is 16 and m + n - q is greater than 4, exit the m loop, since no scheduling
                                    is done after 16:20
                                    If (k = 16 And m + n - q > 4) Then Exit For
                                    ' Check if current block is last block of the current hour and if the consecutive
                                    blocks are less than necessaryblocks

```

```

        If m + n - q > numBlocksPerHour And consecutiveBlocks < necessaryblocks Then
            startk = k
            startm = m
            q = consecutiveBlocks
            ' Continue to next hour
            Exit For
        End If
        If ScheduleCT1LEPT(o, k, m + n - q) = 0 Then
            consecutiveBlocks = consecutiveBlocks + 1
        Else
            consecutiveBlocks = 0
            Exit For
        End If
    Next n

    ' If k is 16 and m + n - q is greater than 4, exit the m loop, since no scheduling is
done after 16:20
    If (k = 16 And m + n - q > 4) Then Exit For
    ' When arrived at the last block of the hour, go to the next hour
    If m + n > numBlocksPerHour And consecutiveBlocks < necessaryblocks Then Exit For

    'If the necessary blocks equal the found blocks, then schedule the scan using the first
hour a scan can be scheduled in if a scan overlaps multiple hours.
    'Otherwise the code would use the first block from the next hour and not schedule
correctly.
    If consecutiveBlocks = necessaryblocks Then
        If startk = 0 Then
            blockStart = ((k - 8) * numBlocksPerHour) + m
        Else
            blockStart = ((startk - 8) * numBlocksPerHour) + startm
        End If
        'Schedule the scan in the chosen blocks
        For p = blockStart To blockStart + necessaryblocks - 1
            ScheduleCT1LEPT(o, 8 + ((p - 1) \ numBlocksPerHour), (p - 1) Mod
numBlocksPerHour + 1) = scanNumber
        Next p
        'Go to the next scan
        GoTo NextStep
    End If
End If
Next m
' If k is 16 and m + n - q is greater than 4, exit the m loop and reset variables
If (k = 16 And m + n - q > 4) Then
    q = 0
    consecutiveBlocks = 0
    r = 0
    k = k - 1
    m = oldm
    startk = 0
    startm = 0
    Exit For
End If
Next k
Next o
NextStep:
Next j

'Realize the LEPT CT1 schedule
For i = 1 To 5
    For j = 8 To 19
        For k = 1 To 12
            scanNumber = ScheduleCT1LEPT(i, j, k)
            r = 0
            If scanNumber = "0" Then Exit For
            For q = LBound(ScansN, 2) To UBound(ScansN, 2)
                If scanNumber = ScansN(1, q) Then
                    r = q
                    Exit For
                End If
            Next q
            duration = ScansN(3, r)
            arrivalDay = ScansN(4, r)
            arrivalHour = ScansN(5, r)
            scheduledduration = ScansN(6, r)
            actualblocks = Application.WorksheetFunction.RoundUp((duration / (numMinutesPerBlock * 60)), 0)

            'If there is a scan scheduled in the block and the scan is not already scheduled, then perform the
scan in the next available block
            allScans = RetrieveScanNumbers(ScheduleRealizedCT1LEPT)
            matchIndex = Application.Match(scanNumber, allScans, 0)

```



```

        If (scanNumber <> "0" And IsError(matchIndex)) Then
            For m = 1 To 5
                For n = 8 To 24
                    For o = 1 To 12
                        If (ScheduleRealizedCT1LEPT(m, n, o) = 0 And m >= i And ((n - 8) * 12 + o) >=
(((j - 8) * 12) + k)) Then
                            blockStart = (((n - 8) * 12) + o)
                            For p = blockStart To blockStart + actualblocks - 1
                                ScheduleRealizedCT1LEPT(m, 8 + ((p - 1) \ numBlocksPerHour), (p - 1) Mod
numBlocksPerHour + 1) = scanNumber
                            Next p
                            GoTo SkipCT1LEPT
                        End If
                    Next o
                Next n
            Next m
        End If
SkipCT1LEPT:
        Next k
        If scanNumber = "0" Then Exit For
    Next j
Next i

'SEPT
' Sort the array based on the expected scantime and system arrival time
Call Sort2DArrayAsc(ScansN, Array(6, 4, 5), True)

'Iterate over the generated scans
For j = LBound(ScansN, 2) To LastNonZeroColumn(ScansN)
    'Get the scan information
    scanNumber = ScansN(1, j)
    duration = ScansN(3, j)
    arrivalDay = ScansN(4, j)
    arrivalHour = ScansN(5, j)
    scheduledduration = ScansN(6, j)
    necessaryblocks = Application.WorksheetFunction.RoundUp((scheduledduration / (numMinutesPerBlock * 60)), 0)
    actualblocks = Application.WorksheetFunction.RoundUp((duration / (numMinutesPerBlock * 60)), 0)

    'Find the next available block in the schedule and reset the variables
    consecutiveBlocks = 0
    q = 0
    r = 0
    startk = 0
    startm = 0

    For o = 1 To 5
        For k = 8 To 16
            For m = 1 To numBlocksPerHour
                'If k is 16 and m is greater than 4, exit the m loop
                If (k = 16 And m > 4) Then Exit For
                If (ReservedCTSEH(o, k, m) = 1 And ScheduleCTSEHSEPT(o, k, m) = 0 And necessaryblocks <= 3)
Then
                    For n = consecutiveBlocks To necessaryblocks - 1
                        'If k is 16 and m + n - q is greater than 4, exit the m loop, since no scheduling
is done after 16:20
                            If (k = 16 And m + n - q > 4) Then Exit For
                        'Search only if the schedule is empty, not reserved and a scan does not take longer than 4
blocks
                            If m + n - q > numBlocksPerHour And consecutiveBlocks < necessaryblocks Then
                                q = consecutiveBlocks
                                r = 1
                                oldm = m
                                startk = k
                                startm = m
                                'Continue to next hour and remember the start hour and block for the scheduling
                                Exit For
                            End If
                            'If the block is reserved and the schedule is empty, remember how many blocks are
found and look for the next block
                            If ReservedCTSEH(o, k, m + n - q) = 1 And ScheduleCTSEHSEPT(o, k, m + n - q) = 0
Then
                                consecutiveBlocks = consecutiveBlocks + 1
                            Else
                                consecutiveBlocks = 0
                                Exit For
                            End If
                        Next n

                        'If k is 16 and m + n - q is greater than 4, exit the m loop, since no scheduling
is done after 16:20
                        If (k = 16 And m + n - q > 4) Then Exit For
                    
```

```

        ' When arrived at the last block of the hour, go to the next hour
        If m + n > numBlocksPerHour And consecutiveBlocks < necessaryblocks Then Exit For

        'If the necessary blocks equal the found blocks, then schedule the scan using the first
hour a scan can be scheduled in if a scan overlaps multiple hours.
        'Otherwise the code would use the first block from the next hour and not schedule
correctly.

        If consecutiveBlocks = necessaryblocks Then
            If startk = 0 Then
                blockStart = ((k - 8) * numBlocksPerHour) + m
            Else
                blockStart = ((startk - 8) * numBlocksPerHour) + startm
            End If
            'Schedule the scan in the chosen blocks
            For p = blockStart To blockStart + necessaryblocks - 1
                ScheduleCTSEHSEPT(o, 8 + ((p - 1) \ numBlocksPerHour), (p - 1) Mod
numBlocksPerHour + 1) = scanNumber
            Next p
            'Go to next scan
            GoTo NextStep2
        End If

    Else
        'If value not found on the CTSEH machine, then reset values and continue the search in
the schedule for CT1

        If r = 1 Then
            q = 0
            consecutiveBlocks = 0
            r = 0
            k = k - 1
            m = oldm
            startk = 0
            startm = 0
        End If
        If ScheduleCT1SEPT(o, k, m) = 0 Then
            'Look in ScheduleCT1
            For n = consecutiveBlocks To necessaryblocks - 1
                'If k is 16 and m + n - q is greater than 4, exit the m loop, since no scheduling
is done after 16:20

                If (k = 16 And m + n - q > 4) Then Exit For
                'Check if current block is last block of current hour and consecutive blocks are
less than duration

                If m + n - q > numBlocksPerHour And consecutiveBlocks < necessaryblocks Then
                    startk = k
                    startm = m
                    q = consecutiveBlocks
                    'Continue to next hour
                    Exit For
                End If
                If ScheduleCT1SEPT(o, k, m + n - q) = 0 Then
                    consecutiveBlocks = consecutiveBlocks + 1
                Else
                    consecutiveBlocks = 0
                    Exit For
                End If
            Next n

            'If k is 16 and m + n - q is greater than 4, exit the m loop, since no scheduling is
done after 16:20

            If (k = 16 And m + n - q > 4) Then Exit For
            ' When arrived at the last block of the hour, go to the next hour
            If m + n > numBlocksPerHour And consecutiveBlocks < necessaryblocks Then Exit For

            'If the necessary blocks equal the found blocks, then schedule the scan using the first
hour a scan can be scheduled in if a scan overlaps multiple hours.
            'Otherwise the code would use the first block from the next hour and not schedule
correctly.

            If consecutiveBlocks = necessaryblocks Then
                If startk = 0 Then
                    blockStart = ((k - 8) * numBlocksPerHour) + m
                Else
                    blockStart = ((startk - 8) * numBlocksPerHour) + startm
                End If
                'Schedule the scan in the chosen blocks
                For p = blockStart To blockStart + necessaryblocks - 1
                    ScheduleCT1SEPT(o, 8 + ((p - 1) \ numBlocksPerHour), (p - 1) Mod
numBlocksPerHour + 1) = scanNumber
                Next p

                'Immediately calculate the realized schedule
                firstIndex = o
                lastNonZeroIndices = FindLastNonZeroIndices(ScheduleRealizedCT1SEPT, firstIndex)

```

```

        If IsError(lastNonZeroIndices) Then
            blockStart = 1
        Else
            k = lastNonZeroIndices(0)
            m = lastNonZeroIndices(1)
            blockStart = ((k - 8) * numBlocksPerHour) + m + 1
        End If
        For p = blockStart To blockStart + actualblocks - 1
            ScheduleRealizedCT1SEPT(o, 8 + ((p - 1) \ numBlocksPerHour), (p - 1) Mod
numBlocksPerHour + 1) = scanNumber
        Next p

        'Go to next scan
        GoTo NextStep2
    End If
End If
End If
Next m
'If k is 16 and m + n - q is greater than 4, exit the m loop and reset variables
If (k = 16 And m + n - q > 4) Then
    q = 0
    consecutiveBlocks = 0
    r = 0
    k = k - 1
    m = oldm
    startk = 0
    startm = 0
    Exit For
End If
Next k
Next o
NextStep2:
Next j

'Realize the SEPT CT1 schedule
For i = 1 To 5
    For j = 8 To 19
        For k = 1 To 12
            scanNumber = ScheduleCT1SEPT(i, j, k)
            r = 0
            If scanNumber = "0" Then Exit For
            For q = LBound(ScansN, 2) To UBound(ScansN, 2)
                If scanNumber = ScansN(1, q) Then
                    r = q
                    Exit For
                End If
            Next q
            duration = ScansN(3, r)
            arrivalDay = ScansN(4, r)
            arrivalHour = ScansN(5, r)
            scheduledduration = ScansN(6, r)
            actualblocks = Application.WorksheetFunction.RoundUp((duration / (numMinutesPerBlock * 60)), 0)

            'If there is a scan scheduled in the block and the scan is not already scheduled, then perform the
            scan in the next available block
            allScans = RetrieveScanNumbers(ScheduleRealizedCT1SEPT)
            matchIndex = Application.Match(scanNumber, allScans, 0)
            If (scanNumber <> "0" And IsError(matchIndex)) Then
                For m = 1 To 5
                    For n = 8 To 24
                        For o = 1 To 12
                            If (ScheduleRealizedCT1SEPT(m, n, o) = 0 And m >= i And ((n - 8) * 12) + o >=
(((j - 8) * 12) + k)) Then
                                blockStart = ((n - 8) * 12) + o
                                For p = blockStart To blockStart + actualblocks - 1
                                    ScheduleRealizedCT1SEPT(m, 8 + ((p - 1) \ numBlocksPerHour), (p - 1) Mod
numBlocksPerHour + 1) = scanNumber
                                Next p
                                GoTo SkipCT1SEPT
                            End If
                        Next o
                    Next n
                Next m
            End If
        Next k
        If scanNumber = "0" Then Exit For
    Next j
Next i

'CTSEH

```

```

'Iterate over the generated scans
For j = LBound(ScansU, 2) To LastNonZeroColumn(ScansU)
    'Get the scan information
    scanNumber = ScansU(1, j)
    duration = ScansU(3, j)
    arrivalDay = ScansU(4, j)
    arrivalHour = ScansU(5, j)
    necessaryblocks = Application.WorksheetFunction.RoundUp((duration / (numMinutesPerBlock * 60)), 0)

    'Find the next available block in the schedule
    For k = arrivalHour To 19
        For m = 1 To numBlocksPerHour
            'But only if the schedule is empty and not reserved
            If ScheduleCTSEH(arrivalDay, k, m) = 0 And ReservedCTSEH(arrivalDay, k, m) = 0 Then
                consecutiveBlocks = 0
                For n = 0 To necessaryblocks - 1
                    If ScheduleCTSEH(arrivalDay, k, m + n) = 0 And ReservedCTSEH(arrivalDay, k, m + n) = 0
Then
                        consecutiveBlocks = consecutiveBlocks + 1
                        ' Check if current block is last block of the current hour and if the
consecutive blocks are less than necessaryblocks
                        If m + n = numBlocksPerHour And consecutiveBlocks < necessaryblocks Then
                            ' Continue to next hour
                            Exit For
                        End If
                    'If the block is not empty or reserved, then go to the next block
                    Else
                        consecutiveBlocks = 0
                        Exit For
                    End If
                Next n
                'If the necessary blocks equal the found blocks, then schedule the scan
                If consecutiveBlocks = necessaryblocks Then
                    blockStart = ((k - 8) * numBlocksPerHour) + m
                    GoTo Schedule
                End If
            End If
        Next m
    Next k

Schedule:

    'Schedule the scan in the chosen blocks
    For k = blockStart To blockStart + (duration \ (numMinutesPerBlock * 60)) - 1
        ScheduleCTSEH(arrivalDay, 8 + ((k - 1) \ numBlocksPerHour), (k - 1) Mod numBlocksPerHour + 1) =
scanNumber
        ScheduleCTSEHLEPT(arrivalDay, 8 + ((k - 1) \ numBlocksPerHour), (k - 1) Mod numBlocksPerHour + 1) =
scanNumber
        ScheduleCTSEHSEPT(arrivalDay, 8 + ((k - 1) \ numBlocksPerHour), (k - 1) Mod numBlocksPerHour + 1) =
scanNumber
    Next k

    'Print the waiting time per patient
    Worksheets("Simulation Output WT").Cells(j + 4, It) = Int((8 + ((blockStart - 1) / numBlocksPerHour)) -
arrivalHour)
Next j

'Realize the CTSEH schedules
'CTSEH LEPT
For i = 1 To 5
    For j = 8 To 19
        For k = 1 To 12
            scanNumber = ScheduleCTSEHLEPT(i, j, k)
            r = 0
            If Left(scanNumber, 1) = "N" Then
                For q = LBound(ScansN, 2) To UBound(ScansN, 2)
                    If scanNumber = ScansN(1, q) Then
                        r = q
                        Exit For
                    End If
                Next q
                duration = ScansN(3, r)
                arrivalDay = ScansN(4, r)
                arrivalHour = ScansN(5, r)
                scheduledduration = ScansN(6, r)
                actualblocks = Application.WorksheetFunction.RoundUp((duration / (numMinutesPerBlock * 60)), 0)
            ElseIf Left(scanNumber, 1) = "U" Then
                For q = LBound(ScansU, 2) To UBound(ScansU, 2)
                    If scanNumber = ScansU(1, q) Then
                        r = q
                        Exit For
                    End If
                Next q
            End If
        Next k
    Next j
Next i

```

```

        End If
    Next q
    duration = ScansU(3, r)
    arrivalDay = ScansU(4, r)
    arrivalHour = ScansU(5, r)
    actualblocks = Application.WorksheetFunction.RoundUp((duration / (numMinutesPerBlock * 60)), 0)
End If

'If there is a scan scheduled in the block and the scan is not already scheduled, then perform the
scan in the next available block
allScans = RetrieveScanNumbers(ScheduleRealizedCTSEHLEPT)
matchIndex = Application.Match(scanNumber, allScans, 0)
If (scanNumber <> "0" And IsError(matchIndex)) Then
    For m = 1 To 5
        For n = 8 To 24
            For o = 1 To 12
                If (ScheduleRealizedCTSEHLEPT(m, n, o) = 0 And m >= i And ((Left(scanNumber, 1) =
"N" And (((n - 8) * 12) + o) >= (((j - 8) * 12) + k)) Or (Left(scanNumber, 1) = "U" And (((n - 8) * 12) + o) >=
((arrivalHour - 8) * 12) + 1))) Then
                    blockStart = (((n - 8) * 12) + o)
                    For p = blockStart To blockStart + actualblocks - 1
                        ScheduleRealizedCTSEHLEPT(m, 8 + ((p - 1) \ numBlocksPerHour), (p - 1) Mod
numBlocksPerHour + 1) = scanNumber
                    Next p
                    GoTo SkipSEH
                End If
            Next o
        Next n
    Next m
End If

SkipSEH:
    Next k
    Next j
Next i

'CTSEH SEPT
For i = 1 To 5
    For j = 8 To 19
        For k = 1 To 12
            scanNumber = ScheduleCTSEHSEPT(i, j, k)
            r = 0
            If Left(scanNumber, 1) = "N" Then
                For q = LBound(ScansN, 2) To UBound(ScansN, 2)
                    If scanNumber = ScansN(1, q) Then
                        r = q
                        Exit For
                    End If
                Next q
                duration = ScansN(3, r)
                arrivalDay = ScansN(4, r)
                arrivalHour = ScansN(5, r)
                scheduledduration = ScansN(6, r)
                actualblocks = Application.WorksheetFunction.RoundUp((duration / (numMinutesPerBlock * 60)), 0)
            ElseIf Left(scanNumber, 1) = "U" Then
                For q = LBound(ScansU, 2) To UBound(ScansU, 2)
                    If scanNumber = ScansU(1, q) Then
                        r = q
                        Exit For
                    End If
                Next q
                duration = ScansU(3, r)
                arrivalDay = ScansU(4, r)
                arrivalHour = ScansU(5, r)
                actualblocks = Application.WorksheetFunction.RoundUp((duration / (numMinutesPerBlock * 60)), 0)
            End If

'If there is a scan scheduled in the block and the scan is not already scheduled, then perform the
scan in the next available block
allScans = RetrieveScanNumbers(ScheduleRealizedCTSEHSEPT)
matchIndex = Application.Match(scanNumber, allScans, 0)
If (scanNumber <> "0" And IsError(matchIndex)) Then
    For m = 1 To 5
        For n = 8 To 24
            For o = 1 To 12
                If (ScheduleRealizedCTSEHSEPT(m, n, o) = 0 And m >= i And ((Left(scanNumber, 1) =
"N" And (((n - 8) * 12) + o) >= (((j - 8) * 12) + k)) Or (Left(scanNumber, 1) = "U" And (((n - 8) * 12) + o) >=
((arrivalHour - 8) * 12) + 1))) Then
                    blockStart = (((n - 8) * 12) + o)
                    For p = blockStart To blockStart + actualblocks - 1
                        ScheduleRealizedCTSEHSEPT(m, 8 + ((p - 1) \ numBlocksPerHour), (p - 1) Mod
numBlocksPerHour + 1) = scanNumber
                    Next p
                End If
            Next o
        Next n
    Next m
End If

```

```

        Next p
        GoTo SkipSEH2
    End If
    Next o
    Next n
    Next m
End If

SkipSEH2:
    Next k
    Next j
Next i

'Print the schedule for each replication
lastnonzerovar = 0
Cmax = 0
For j = 1 To numDays
    If It = wks.Cells(7, 4) Then
        For k = 8 To 24
            For m = 1 To numBlocksPerHour
                If k <= 18 Then
                    wks.Cells(k * numBlocksPerHour + m - 84, j + 20).Value = ScheduleCT1LEPT(j, k, m)
                    wks.Cells(k * numBlocksPerHour + m - 84, j + 46).Value = ScheduleCT1SEPT(j, k, m)
                    Worksheets("Schedule vs. Output").Cells(k * numBlocksPerHour + m - 91, j + 3).Value =
ScheduleCT1LEPT(j, k, m)
                    Worksheets("Schedule vs. Output").Cells(k * numBlocksPerHour + m - 91, j + 35).Value =
ScheduleCT1SEPT(j, k, m)
                End If
                If k <= 19 Then
                    wks.Cells(k * numBlocksPerHour + m - 84, j + 28).Value = ScheduleCTSEH(j, k, m)
                    wks.Cells(k * numBlocksPerHour + m + 51, j + 20).Value = ScheduleCTSEHLEPT(j, k, m)
                    wks.Cells(k * numBlocksPerHour + m + 51, j + 46).Value = ScheduleCTSEHSEPT(j, k, m)
                    Worksheets("Schedule vs. Output").Cells(k * numBlocksPerHour + m - 91, j + 19).Value =
ScheduleCTSEHLEPT(j, k, m)
                    Worksheets("Schedule vs. Output").Cells(k * numBlocksPerHour + m - 91, j + 51).Value =
ScheduleCTSEHSEPT(j, k, m)
                End If
                Worksheets("Schedule vs. Output").Cells(k * numBlocksPerHour + m - 91, j + 11).Value =
ScheduleRealizedCT1LEPT(j, k, m)
                Worksheets("Schedule vs. Output").Cells(k * numBlocksPerHour + m - 91, j + 43).Value =
ScheduleRealizedCT1SEPT(j, k, m)
                Worksheets("Schedule vs. Output").Cells(k * numBlocksPerHour + m - 91, j + 27).Value =
ScheduleRealizedCTSEHLEPT(j, k, m)
                Worksheets("Schedule vs. Output").Cells(k * numBlocksPerHour + m - 91, j + 59).Value =
ScheduleRealizedCTSEHSEPT(j, k, m)
            Next m
        Next k
    End If

    'Print LEPT And SEPT; completion times
    wks.Cells(3 + j, 19) = j
    wks.Cells(3 + j, 20) = CountUniqueNonZero(ScheduleCT1LEPT, j)
    lastnonzerovar = lastNonZero(ScheduleCT1LEPT, j)
    ' If lastnonzerovar = 0 Then
    '     wks.Cells(3 + j, 21) = 0
    ' Else
    '     wks.Cells(3 + j, 21) = lastnonzerovar * ((1 / 12) / 24)
    ' End If

    wks.Cells(3 + j, 24) = j
    wks.Cells(3 + j, 25) = CountUniqueNonZero(ScheduleCT1SEPT, j)
    lastnonzerovar = lastNonZero(ScheduleCT1SEPT, j)
    ' If lastnonzerovar = 0 Then
    '     wks.Cells(3 + j, 26) = 0
    ' Else
    '     wks.Cells(3 + j, 26) = lastnonzerovar * ((1 / 12) / 24)
    ' End If

    lastnonzerovar = lastNonZero(ScheduleCT1LEPT, j)
    Cmax = Cmax + (lastnonzerovar * ((1 / 12) / 24))
    Worksheets("Schedule vs. Output").Cells(2, 8) = Cmax

    lastnonzerovar = lastNonZero(ScheduleRealizedCT1LEPT, j)
    Cmax2 = Cmax2 + (lastnonzerovar * ((1 / 12) / 24))
    Worksheets("Schedule vs. Output").Cells(2, 16) = Cmax2

    lastnonzerovar = lastNonZero(ScheduleCTSEHLEPT, j)
    Cmax3 = Cmax3 + (lastnonzerovar * ((1 / 12) / 24))
    Worksheets("Schedule vs. Output").Cells(2, 24) = Cmax3

    lastnonzerovar = lastNonZero(ScheduleRealizedCTSEHLEPT, j)
    Cmax4 = Cmax4 + (lastnonzerovar * ((1 / 12) / 24))

```

```

Worksheets("Schedule vs. Output").Cells(2, 32) = Cmax4

lastnonzerovar = lastNonZero(ScheduleCT1SEPT, j)
Cmax5 = Cmax5 + (lastnonzerovar * ((1 / 12) / 24))
Worksheets("Schedule vs. Output").Cells(2, 40) = Cmax5

lastnonzerovar = lastNonZero(ScheduleRealizedCT1SEPT, j)
Cmax6 = Cmax6 + (lastnonzerovar * ((1 / 12) / 24))
Worksheets("Schedule vs. Output").Cells(2, 48) = Cmax6

lastnonzerovar = lastNonZero(ScheduleCTSEHSEPT, j)
Cmax7 = Cmax7 + (lastnonzerovar * ((1 / 12) / 24))
Worksheets("Schedule vs. Output").Cells(2, 56) = Cmax7

lastnonzerovar = lastNonZero(ScheduleRealizedCTSEHSEPT, j)
Cmax8 = Cmax8 + (lastnonzerovar * ((1 / 12) / 24))
Worksheets("Schedule vs. Output").Cells(2, 64) = Cmax8
Next j

'Print completion times of scheduled versus output
Worksheets("Simulation Output CMAX").Cells(5, It) = Cmax2
Worksheets("Simulation Output CMAX").Cells(12, It) = Cmax6

'Add border to data
wks.Range("S4:V8").BorderAround LineStyle:=xlContinuous, Weight:=xlThin
wks.Range("X4:AA8").BorderAround LineStyle:=xlContinuous, Weight:=xlThin

End Sub

Function lastNonZero(arr() As Variant, ByVal j As Integer) As Long
    Dim i, k As Integer

    'Find the lastnonzero value in the array to find out when the last scan is scheduled
    For i = UBound(arr, 2) To LBound(arr, 2) Step -1
        For k = UBound(arr, 3) To LBound(arr, 3) Step -1
            If arr(j, i, k) <> 0 Then
                lastNonZero = (12 * (i - 8)) + k
                Exit Function
            End If
        Next k
    Next i

End Function

Function CountUniqueNonZero(arr As Variant, ByVal j As Integer) As Long
    Dim i As Integer, k As Integer
    Dim dict As Object
    Set dict = CreateObject("Scripting.Dictionary")

    'Count the unique non zero values in an array to find out how many scans are scheduled
    For i = LBound(arr, 2) To UBound(arr, 2)
        For k = LBound(arr, 3) To UBound(arr, 3)
            If arr(j, i, k) <> 0 Then
                If Not dict.Exists(arr(j, i, k)) Then
                    dict.Add arr(j, i, k), 1
                End If
            End If
        Next k
    Next i

    CountUniqueNonZero = dict.Count
End Function

Function LastNonZeroColumn(arr As Variant) As Long
    Dim i As Long, j As Long, lastCol As Long

    lastCol = 0

    'Find the last non zero column to clear all values in the column
    For j = UBound(arr, 2) To 1 Step -1
        If arr(1, j) <> 0 Then
            lastCol = j
            Exit For
        End If
    End If
    If lastCol <> 0 Then Exit For
Next j

LastNonZeroColumn = lastCol
End Function

Function Sort2DArrayDesc(arr As Variant) As Variant

```

```

Dim i As Long, j As Long, k As Long
Dim temp As Variant

For i = LBound(arr, 2) To UBound(arr, 2) - 1
    For j = i + 1 To UBound(arr, 2)
        If arr(6, i) < arr(6, j) Then
            For k = LBound(arr, 1) To UBound(arr, 1)
                temp = arr(k, i)
                arr(k, i) = arr(k, j)
                arr(k, j) = temp
            Next k
        ElseIf arr(6, i) = arr(6, j) Then
            If arr(4, i) > arr(4, j) Then
                For k = LBound(arr, 1) To UBound(arr, 1)
                    temp = arr(k, i)
                    arr(k, i) = arr(k, j)
                    arr(k, j) = temp
                Next k
            ElseIf arr(4, i) = arr(4, j) And arr(5, i) > arr(5, j) Then
                For k = LBound(arr, 1) To UBound(arr, 1)
                    temp = arr(k, i)
                    arr(k, i) = arr(k, j)
                    arr(k, j) = temp
                Next k
            End If
        End If
    Next j
Next i

Sort2DArrayDesc = arr
End Function

Sub Sort2DArrayAsc(ByRef arr As Variant, ByVal sortColumns As Variant, ByVal ascending As Boolean)
    Dim temp As Variant
    Dim i As Long, j As Long, k As Long
    Dim shouldSwap As Boolean

    ' Bubble sort the array based on the values in the specified columns
    For i = UBound(arr, 2) To 2 Step -1
        For j = 1 To i - 1
            shouldSwap = False
            If (arr(sortColumns(0), j) > arr(sortColumns(0), j + 1)) Xor Not ascending Then
                shouldSwap = True
            ElseIf (arr(sortColumns(0), j) = arr(sortColumns(0), j + 1)) Then
                If (arr(sortColumns(1), j) > arr(sortColumns(1), j + 1)) Xor Not ascending Then
                    shouldSwap = True
                ElseIf (arr(sortColumns(1), j) = arr(sortColumns(1), j + 1)) Then
                    If (arr(sortColumns(2), j) > arr(sortColumns(2), j + 1)) Xor Not ascending Then
                        shouldSwap = True
                    End If
                End If
            End If
            If shouldSwap Then
                For k = 1 To UBound(arr, 1)
                    temp = arr(k, j)
                    arr(k, j) = arr(k, j + 1)
                    arr(k, j + 1) = temp
                Next k
            End If
        Next j
    Next i
End Sub

Function RetrieveScanNumbers(inputArray As Variant) As Variant
    'Create a Dictionary object to store unique values
    Dim uniqueValues As Object
    Set uniqueValues = CreateObject("Scripting.Dictionary")

    'Loop through every element of the 3D array and add unique values to the Dictionary
    Dim i As Long, j As Long, k As Long
    For i = LBound(inputArray, 1) To UBound(inputArray, 1)
        For j = LBound(inputArray, 2) To UBound(inputArray, 2)
            For k = LBound(inputArray, 3) To UBound(inputArray, 3)
                If Not uniqueValues.exists(inputArray(i, j, k)) Then
                    uniqueValues.Add inputArray(i, j, k), 0
                End If
            Next k
        Next j
    Next i

    'Create a 1D array from the keys of the Dictionary object

```



```

Dim uniqueArray() As Variant
ReDim uniqueArray(0 To uniqueValues.Count - 1)

Dim key As Variant, index As Long
For Each key In uniqueValues.keys()
    uniqueArray(index) = key
    index = index + 1
Next key

'Return the 1D array
RetrieveScanNumbers = uniqueArray
End Function

Function FindLastNonZeroIndices(inputArray As Variant, firstIndex As Long) As Variant
'Loop through the 3D array in reverse order and return the indices of the last non-zero value
Dim j As Long, k As Long
For j = UBound(inputArray, 2) To LBound(inputArray, 2) Step -1
    For k = UBound(inputArray, 3) To LBound(inputArray, 3) Step -1
        If inputArray(firstIndex, j, k) <> 0 Then
            FindLastNonZeroIndices = Array(j, k)
            Exit Function
        End If
    Next k
Next j

'If no non-zero value is found, return an error value
FindLastNonZeroIndices = CVErr(xlErrNA)
End Function

```

Appendix D.4: Experiments VBA Code

```

Sub RunSim()

Dim i, j, k, m, NrofIt, NrofScans, lastColumn, lastRow, It As Integer
Dim wks As Excel.Worksheet
Dim ScheduleblockSEH(1 To 5, 1 To 144) As Integer
Dim ReplicationLEPT As Variant
Dim ReplicationSEPT As Variant
Dim TotalCMAXLEPT As Variant
Dim TotalCMAXSEPT As Variant
Dim TotalLEPT As Variant
Dim TotalSEPT As Variant
Dim Cmax(1 To 8) As Date

Dim StartTime As Double
Dim SecondsElapsed As Double

'Remember time when macro starts
StartTime = Timer

Set wks = ThisWorkbook.Worksheets("Monte Carlo Simulation")

Application.Calculation = xlCalculationManual
Application.ScreenUpdating = False

'Clear simulation output from last simulation
lastColumn = Worksheets("Simulation Output WT").Cells(5, Columns.Count).End(xlToLeft).Column
Worksheets("Simulation Output WT").Range(Worksheets("Simulation Output WT").Cells(5, 1), Worksheets("Simulation
Output WT").Cells(400, lastColumn)).ClearContents
lastColumn = Worksheets("Simulation Output CMAX").Cells(5, Columns.Count).End(xlToLeft).Column
Worksheets("Simulation Output CMAX").Range(Worksheets("Simulation Output CMAX").Cells(5, 1),
Worksheets("Simulation Output CMAX").Cells(5, lastColumn)).ClearContents
Worksheets("Simulation Output CMAX").Range(Worksheets("Simulation Output CMAX").Cells(12, 1),
Worksheets("Simulation Output CMAX").Cells(12, lastColumn)).ClearContents
wks.Range("S4:AA8").ClearContents

' # Of Replications from sheet as input
NrofIt = wks.Cells(7, 4)

' ReDim for a different outputs every Replication
ReDim ReplicationLEPT(1 To NrofIt, 1 To 5), ReplicationSEPT(1 To NrofIt, 1 To 5) As Variant
ReDim TotalLEPT(1 To NrofIt), TotalSEPT(1 To NrofIt) As Variant

' Clear arrays
For j = 1 To 5
    For k = 1 To 132
        ScheduleblockSEH(j, k) = 0
    Next k
Next j

For j = 1 To 5

```

```

    For k = 8 To 16
        For m = 1 To 12
            ReservedCTSEHBASIC(j, k, m) = 0
        Next m
    Next k
Next j

'Sim over # of Replications
For It = 1 To NrofIt
    Call ScanGeneration(It)
    Call ScheduleScans(It)
    For j = 1 To 5
        For k = 1 To 144
            If ScheduleCTSEH(j, 8 + Int(k / 12), k Mod 12) = 0 Then
                ScheduleblockSEH(j, k) = ScheduleblockSEH(j, k) + 1
            End If
        Next k

        'Retrieve replication makespan
        ReplicationLEPT(It, j) = wks.Cells(3 + j, 21)
        ReplicationSEPT(It, j) = wks.Cells(3 + j, 26)

        'Add Replication to total
        TotalLEPT(It) = TotalLEPT(It) + ReplicationLEPT(It, j)
        TotalSEPT(It) = TotalSEPT(It) + ReplicationSEPT(It, j)

    Next j

    'Sum and print total of every iteration
    TotalCMAXLEPT = TotalCMAXLEPT + TotalLEPT(It)
    wks.Cells(13, 4) = TotalLEPT(It)
    '    Worksheets("Simulation Output CMAX").Cells(5, It) = TotalLEPT(It)
    TotalCMAXSEPT = TotalCMAXSEPT + TotalSEPT(It)
    wks.Cells(17, 4) = TotalSEPT(It)
    '    Worksheets("Simulation Output CMAX").Cells(12, It) = TotalSEPT(It)

    'Retrieve all Cmax values from schedule versus output
    For i = 1 To 8
        Cmax(i) = Cmax(i) + Worksheets("Schedule vs. Output").Cells(2, 8 * i)
    Next i

Next It

'Calculate and print the average percentage of empty CTSEH cells
For j = 1 To 5
    For k = 1 To 144
        wks.Cells(k + 12, j + 36) = ScheduleblockSEH(j, k) / NrofIt
    Next k
Next j

'Print Timer
wks.Cells(3, 29) = Round(Timer - StartTime, 2)
wks.Cells(4, 29) = Round(Timer - StartTime, 2) / NrofIt

'Retrieve Total Average per Replication
wks.Cells(14, 4) = TotalCMAXLEPT / NrofIt
wks.Cells(18, 4) = TotalCMAXSEPT / NrofIt

'Print total Cmax values of schedule versus output
For i = 1 To 8
    Worksheets("Schedule vs. Output").Cells(2, 8 * i) = Cmax(i) / NrofIt
Next i

Application.Calculation = xlCalculationAutomatic
Application.ScreenUpdating = True

'Print reserved timeslots, waiting time and makespan
lastRow = ThisWorkbook.Worksheets("SO RTS").Cells(Rows.Count, 1).End(xlUp).Row
wks.Range("AJ159:A0168").Copy (Worksheets("SO RTS").Range("A" & lastRow + 2))
Worksheets("SO RTS").Cells(lastRow + 2, 8) = Worksheets("Simulation Output WT").Cells(1, 8)
Worksheets("SO RTS").Cells(lastRow + 2, 9) = Worksheets("Simulation Output WT").Cells(1, 5)
Worksheets("SO RTS").Cells(lastRow + 2, 10) = Worksheets("Simulation Output WT").Cells(2, 5)
Worksheets("SO RTS").Cells(lastRow + 2, 12) = Worksheets("Simulation Output CMAX").Cells(1, 8)
Worksheets("SO RTS").Cells(lastRow + 2, 13) = Worksheets("Simulation Output CMAX").Cells(1, 5)
Worksheets("SO RTS").Cells(lastRow + 2, 14) = Worksheets("Simulation Output CMAX").Cells(2, 5)
Worksheets("SO RTS").Cells(lastRow + 2, 16) = Worksheets("Simulation Output CMAX").Cells(8, 8)
Worksheets("SO RTS").Cells(lastRow + 2, 17) = Worksheets("Simulation Output CMAX").Cells(8, 5)
Worksheets("SO RTS").Cells(lastRow + 2, 18) = Worksheets("Simulation Output CMAX").Cells(9, 5)

End Sub

```

APPENDIX E: SIMULATION INPUT

Appendix E.1: Scan types and their occurrence rate

Table 10: Scan types and occurrence rates from CT1 in 2019, Total = 7089 (MST, 2022)

C-Code	Occurance	C-Code	Occurance	C-Code	Occurance
C7-71E	21.40%	C7-72D	0.31%	C5-59X	0.06%
C7-71D	8.96%	C9-92	0.25%	C9-94 B	0.06%
C5-57	8.60%	C7-90	0.25%	C1-16	0.06%
C1-20	6.21%	C1-02C	0.25%	C5-59C	0.06%
C5-59	6.14%	C1-30	0.25%	C5-70	0.06%
C5-50	4.15%	C7-72E	0.25%	C5-51	0.06%
C7-70G	3.48%	I6-03	0.24%	C7-71	0.04%
C2-21	3.37%	C7-71C	0.24%	C2-23	0.04%
C7-72A	3.12%	C1-07	0.23%	C4-46 L	0.04%
C5-61	2.84%	C5-80	0.21%	C4-42 L	0.04%
C5-59A	2.44%	C5-50E	0.18%	C7-70A	0.04%
C7-70I	2.29%	C1-01C	0.18%	C2-22	0.04%
C7-90C	2.27%	C5-51A	0.17%	C1-18	0.04%
C7-70N	2.23%	C1-02	0.16%	C9-96 R	0.03%
C5-50C	2.20%	C7-80	0.13%	C7-70E	0.03%
C1-01	1.58%	C5-55	0.13%	C1-03	0.03%
C7-74	1.17%	C7-71A	0.13%	CT-01	0.03%
C7-70J	1.06%	C7-75F	0.11%	C5-50D	0.03%
C2-24	1.06%	C3-33	0.11%	C9-93 L	0.03%
C7-75	1.03%	C1-05	0.11%	C5-81	0.03%
C2-25	0.87%	C5-56	0.10%	C9-93 R	0.03%
C7-72C	0.86%	C4-42 R	0.10%	C4-46 R	0.03%
C5-57N	0.85%	C5-51E	0.10%	C7-72	0.03%
C5-50A	0.83%	C9-96AL	0.08%	C9-92A	0.01%
C7-70D	0.82%	I4-04	0.08%	C9-96 L	0.01%
C7-70H	0.69%	C5-55C	0.08%	C1-14	0.01%
C1-19	0.61%	C7-70	0.07%	C1-17	0.01%
C7-80T	0.49%	C5-54	0.07%	C7-70T	0.01%
C5-54A	0.44%	C5-51F	0.07%	C5-51D	0.01%
C5-51C	0.42%	C5-55T	0.07%	C9-93 B	0.01%
C7-80I	0.38%	C9-94 L	0.07%	C3-33A	0.01%
C3-31	0.38%	C9-94 R	0.07%	C1-01A	0.01%
C1-21	0.35%	C3-32	0.07%	C1-01D	0.01%
C2-30	0.34%	C4-44 R	0.06%	C5-57C	0.01%

Table 11: Scan types and occurrence rates from CTSEH in 2019, Total = 17583 (MST, 2022)

C-Code	Occurance	C-Code	Occurance	C-Code	Occurance
C5-50C	15.52%	C4-42 L	0.36%	C9-93 B	0.07%
C1-01	14.67%	C1-16	0.36%	C9-94 B	0.07%
C7-71D	7.25%	C9-96 R	0.33%	C5-80	0.07%
C2-21	6.63%	C5-55C	0.32%	C5-51E	0.05%
C7-70H	6.38%	C7-72C	0.31%	C1-11	0.05%
C3-31	4.89%	C9-96AR	0.29%	C7-72D	0.05%
C7-71E	4.00%	C4-46 R	0.28%	C7-70J	0.05%
C5-50A	3.13%	C1-19	0.28%	C1-01A	0.04%
C5-54A	2.80%	C1-17	0.28%	C5-57C	0.03%
C1-05	2.32%	C5-51C	0.25%	C1-01T	0.03%
C1-01C	2.30%	C5-51F	0.25%	C4-42 B	0.03%
C5-50	1.94%	C9-96AL	0.24%	C5-55	0.03%
C2-24	1.79%	C1-18	0.22%	C4-46 B	0.03%
C1-20	1.60%	C9-96 L	0.21%	C7-71	0.03%
C5-57	1.50%	C5-56	0.20%	C7-72E	0.03%
C9-92	1.46%	C5-59A	0.18%	CT-03	0.03%
C3-33	1.31%	C9-93 R	0.18%	C2-23	0.02%
C1-30	1.17%	C5-61	0.17%	C7-70A	0.02%
C5-50E	1.15%	C7-70N	0.17%	C5-59C	0.02%
C7-72A	0.99%	C4-44 L	0.16%	C1-03	0.02%
C7-70G	0.96%	C9-93 L	0.16%	C5-51	0.02%
C2-30	0.84%	C7-74	0.16%	C7-75F	0.02%
C7-71C	0.80%	C7-80I	0.15%	C9-96 B	0.02%
C5-54	0.69%	C4-44 R	0.13%	C5-55T	0.02%
C1-02C	0.68%	C1-02	0.13%	CT-05	0.01%
C2-25	0.60%	C2-22	0.13%	C7-70	0.01%
C7-70D	0.60%	C3-33A	0.12%	PH-03	0.01%
C9-94 L	0.56%	C7-70I	0.12%	C7-80T	0.01%
C3-32	0.54%	C5-59	0.09%	CT-04	0.01%
C9-94 R	0.53%	C9-92A	0.09%	C7-90	0.01%
C4-42 R	0.52%	C7-70C	0.09%	I6-03	0.01%
C5-51A	0.48%	C5-50D	0.09%	C1-14	0.01%
C5-57N	0.45%	C1-02A	0.09%	C1-01U	0.01%
C4-46 L	0.43%	C1-21	0.09%	C4-44 B	0.01%
CT-01	0.41%	C5-70	0.08%	C1-01U	0.01%
C7-75	0.38%	C7-80	0.08%		

Appendix E.1: Replications

Table 12: Calculations for replications of maximum completion time

	KPIs	Mean	Variance	T-value	CIHW (**)	Error	Error in %
1	28:30:00	1.1875					
2	26:50:00	1.1528	0.0012	12.7062	0.3120	0.2706	27.06%
3	33:25:00	1.2326	0.0136	4.3027	0.2893	0.2347	23.47%
4	31:50:00	1.2561	0.0118	3.1824	0.1730	0.1377	13.77%
5	24:45:00	1.2111	0.0175	2.7764	0.1645	0.1358	13.58%
...
15	25:55:00	1.2507	0.0168	2.1448	0.0718	0.0574	5.74%
16	27:00:00	1.2428	0.0167	2.1314	0.0688	0.0554	5.54%
17	33:50:00	1.2527	0.0172	2.1199	0.0675	0.0539	5.39%
18	28:45:00	1.2496	0.0164	2.1098	0.0638	0.0510	5.10%
19	31:10:00	1.2522	0.0157	2.1009	0.0604	0.0482	4.82%

(*) The column KPI should contain the results for each replication of one experiment, in this case results of 500 replications.

(**) Confidence Interval Half Width.

Table 13: Calculations for replications of waiting time

	KPIs	Mean	Variance	T-value	CIHW (**)	Error	Error in %
1	0.1043	0.1043	0.0000		0.0000		
2	0.3786	0.2415	0.0188	12.7062	1.2320	5.1021	510.213%
3	0.2016	0.2282	0.0129	4.3027	0.2820	1.2358	123.584%
4	0.2131	0.2244	0.0097	3.1824	0.1568	0.6986	69.862%
5	0.3042	0.2404	0.0088	2.7764	0.1164	0.4841	48.415%
...
370	0.1085	0.1632	0.0064	1.9664	0.0082	0.0503	5.029%
371	0.1422	0.1632	0.0064	1.9664	0.0082	0.0502	5.018%
372	0.1965	0.1633	0.0064	1.9664	0.0082	0.0500	5.002%
373	0.0777	0.1630	0.0064	1.9664	0.0082	0.0500	5.004%
374	0.2128	0.1632	0.0064	1.9663	0.0081	0.0499	4.989%
375	0.1308	0.1631	0.0064	1.9663	0.0081	0.0498	4.979%

(*) The column KPI should contain the results for each replication of one experiment, in this case results of 500 replications.

(**) Confidence Interval Half Width.

Appendix E.2: Reserved slot calculations example

Avg empty CTSEH							Avg empty CTSEH										
Day							Day										
Hour	Block	1	2	3	4	5		Hour	Block	1	2	3	4	5			
08:00	1	0%	0%	0%	0%	0%		08:00	1	0	0	0	0	0			
08:05	2	0%	0%	0%	0%	0%		08:05	2	0	0	0.333333	0.666667	0			
08:10	3	1%	0%	1%	1%	0%		08:10	3	0.333333	0.083333	0.25	0.25	0.083333			
08:15	4	2%	3%	2%	3%	2%		08:15	4	0.166667	0.238095	0.190476	0.238095	0.166667			
08:20	5	4%	5%	5%	4%	3%		08:20	5	0.197531	0.209877	0.246914	0.185185	0.160494			
08:25	6	7%	7%	11%	7%	6%		08:25	6	0.178571	0.185714	0.285714	0.185714	0.164286			
08:30	7	13%	11%	14%	11%	9%		08:30	7	0.228311	0.191781	0.242009	0.187215	0.150685			
08:35	8	20%	17%	20%	15%	15%		08:35	8	0.227414	0.199377	0.227414	0.17134	0.174455			
08:40	9	26%	23%	25%	21%	23%		08:40	9	0.220455	0.193182	0.211364	0.172723	0.197727			
08:45	10	34%	33%	32%	27%	33%		08:45	10	0.214165	0.20742	0.200675	0.172007	0.205734			
08:50	11	45%	42%	41%	37%	43%		08:50	11	0.217784	0.201031	0.198454	0.177835	0.204897			1
08:55	12	59%	57%	57%	52%	56%	16.38%	08:55	12	0.209724	0.202097	0.202097	0.185891	0.200191	8.928877	8	1.80
09:00	13	1%	0%	0%	0%	0%		09:00	13	0.5	0	0.25	0	0.25		9	2.73
09:05	14	1%	0%	1%	0%	0%		09:05	14	0.5	0	0.375	0	0.125		10	1.69
09:10	15	2%	1%	2%	2%	1%		09:10	15	0.230769	0.115385	0.307692	0.230769	0.115385		11	2.46
09:15	16	3%	2%	3%	3%	2%		09:15	16	0.244898	0.142857	0.22449	0.244898	0.142857		12	2.27
09:20	17	4%	4%	5%	6%	4%		09:20	17	0.186047	0.174419	0.197674	0.255814	0.186047		13	4.61
09:25	18	8%	9%	10%	10%	9%		09:25	18	0.183432	0.189349	0.218935	0.218935	0.189349		14	5.74
09:30	19	14%	13%	16%	16%	13%		09:30	19	0.200743	0.174721	0.226766	0.223048	0.174721		15	4.05
09:35	20	24%	18%	23%	22%	19%		09:35	20	0.22335	0.175127	0.215736	0.205584	0.180203		16	1.40

Figure 20: Reserved slot calculation example

The percentages in the left table are the total times the block was empty divided by the number of replications. The 16.38% is the sum of all percentages of hour 8, day 1 to 5, divided by the count of blocks, 60. The right table shows the respective block percentage divided by the sum of percentages of the respective block of all days. The calculation for the number in block 4, day 1 in the right table is $2/(2 + 3 + 2 + 3 + 2) = 0.16667$. This is the respective percentage, 2%, from block 4, day 1 in the left table, divided by the sum of the percentages of block 4, day 1 to 5 from the left table. The value 9.828877 shows the average empty slots percentage (16.38%) multiplied by the total number of slots per hour (60). This value tells us the average amount of empty scheduling blocks per hour spread over 5 days. The utmost right values give us the number of reserved timeslots per hour per day, which is calculated by taking the average of the values per hour per day in the right table multiplied by the number of average empty slots per hour (9.828877). These values are rounded to its nearest integer to find a potential optimal number of reserved slots per hour.

APPENDIX F: SIMULATION OUTPUT

Appendix F.1: Seven evaluated schedules and outcomes

Reserved Timeslots					WT	Avg	Max	Min	CHASLEP	Avg	Max	Min	CHASSEP	Avg	Max	Min
8	0	0	0	0	0											
9	0	0	0	0	0											
10	0	0	0	0	0											
11	0	0	0	0	0											
12	0	0	0	0	0											
13	0	0	0	0	0											
14	0	0	0	0	0											
15	0	0	0	0	0											
16	0	0	0	0	0											
1	1	2	3	4	5	0.230809		2	0	35:33:48	44:40:00	29:10:00	39:55:00	50:10:00	32:55:00	
8	1	1	1	1	1											
9	1	1	1	1	1											
10	1	1	1	1	1											
11	1	1	1	1	1											
12	1	1	1	1	1											
13	1	1	1	1	1											
14	1	1	1	1	1											
15	1	1	1	1	1											
16	1	1	1	1	1											
1	2	3	4	5		0.375609		3	0	35:33:48	41:50:00	30:25:00	40:38:36	48:15:00	35:10:00	
8	2	2	2	2	2											
9	2	2	2	2	2											
10	2	2	2	2	2											
11	2	2	2	2	2											
12	2	2	2	2	2											
13	2	2	2	2	2											
14	2	2	2	2	2											
15	2	2	2	2	2											
16	2	2	2	2	2											
1	3	3	3	3	3	0.375609		5	0	31:56:24	39:40:00	28:05:00	33:42:12	41:05:00	30:05:00	
8	3	3	3	3	3											
9	3	3	3	3	3											
10	3	3	3	3	3											
11	3	3	3	3	3											
12	3	3	3	3	3											
13	3	3	3	3	3											
14	3	3	3	3	3											
15	3	3	3	3	3											
16	3	3	3	3	3											
1	2	3	4	5		0.600175		5	0	29:44:12	37:30:00	24:10:00	35:35:48	43:30:00	28:30:00	
8	2	2	2	2	1											
9	3	2	3	2	2											
10	2	2	2	1	2											
11	2	2	4	3	3											
12	7	7	8	7	7											
13	5	4	5	5	5											
14	6	5	5	5	5											
15	4	3	4	4	3											
16	2	1	1	1	1											
1	1	2	3	4	5	0.263365		3	0	36:23:48	42:40:00	25:15:00	41:36:00	49:05:00	29:50:00	
8	1	1	1	1	1											
9	1	1	1	1	1											
10	1	1	1	1	1											
11	1	1	1	1	1											
12	2	2	2	2	2											
13	2	2	2	2	2											
14	2	2	2	2	2											
15	2	2	2	2	2											
16	0	0	0	0	0											
1	2	3	4	5		0.430984		4	0	33:08:36	40:40:00	28:10:00	38:33:36	46:55:00	33:10:00	
8	2	2	2	2	2											
9	2	2	2	2	2											
10	2	2	2	2	2											
11	2	2	2	2	2											
12	3	3	3	3	3											
13	3	3	3	3	3											
14	3	3	3	3	3											
15	3	3	3	3	3											
16	0	0	0	0	0											

Figure 21: Seven different evaluated schedules and their simulation outcomes

Appendix F.2: T-tests

t-Test: Paired Two Sample for Means	Between LEPT schedule 1 and 5	
	<i>Variable 1</i>	<i>Variable 2</i>
Mean	41:57:36	29:44:12
Variance	0.040036	0.020319
Observations	25	25
Pearson Correlation	0.041374	
Hypothesized Mean Difference	0	
df	24	
t Stat	10.57437	
P(T<=t) one-tail	8.17E-11	
t Critical one-tail	1.710882	
P(T<=t) two-tail	1.63E-10	
t Critical two-tail	2.063899	
t-Test: Paired Two Sample for Means	Between SEPT schedule 1 and 5	
	<i>Variable 1</i>	<i>Variable 2</i>
Mean	41:57:36	35:35:48
Variance	0.040036	0.023752
Observations	25	25
Pearson Correlation	0.053301	
Hypothesized Mean Difference	0	
df	24	
t Stat	5.38969	
P(T<=t) one-tail	7.77E-06	
t Critical one-tail	1.710882	
P(T<=t) two-tail	1.55E-05	
t Critical two-tail	2.063899	
t-Test: Paired Two Sample for Means	Between LEPT and SEPT schedule 5	
	<i>Variable 1</i>	<i>Variable 2</i>
Mean	29:44:12	35:35:48
Variance	0.020319	0.023752
Observations	25	25
Pearson Correlation	0.938821	
Hypothesized Mean Difference	0	
df	24	
t Stat	-22.9819	
P(T<=t) one-tail	3.74E-18	
t Critical one-tail	1.710882	
P(T<=t) two-tail	7.48E-18	
t Critical two-tail	2.063899	

Figure 22: T-tests

Appendix F.3: Urgent patient waiting time examples.

Waiting Time Urgent Patients		Max	5	Avg	0.568624											
Hours per scan		Min	0													
Replication 1	Replication 2	Replication 3	...													
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	1	0
0	0	0	0	1	1	0	0	0	0	0	1	0	1	1	1	1
0	0	0	0	1	1	0	0	0	0	0	1	1	1	0	1	1
0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	1	0
0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	0
0	0	0	0	1	2	0	1	0	0	0	1	1	1	1	1	0
0	0	0	0	1	2	0	0	0	0	0	1	0	1	2	1	0
0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	1	0
0	0	0	1	1	1	0	0	0	0	0	1	2	1	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	2	1	1	1	1
0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	1	1
0	0	1	0	0	0	0	0	0	0	0	3	0	1	2	0	1
0	1	1	0	0	0	0	0	0	0	0	0	1	1	2	1	1
0	0	1	0	1	0	0	0	0	0	0	1	1	1	2	1	1
0	0	0	0	1	0	0	0	0	0	0	0	1	2	1	1	0
0	0	0	0	1	1	0	0	1	1	0	1	1	2	1	1	1
0	0	0	1	0	1	0	0	1	0	0	0	2	3	1	1	1
0	0	2	0	1	1	0	0	1	0	1	1	1	1	2	2	1
0	0	0	0	2	0	0	0	0	1	0	0	3	2	1	1	1
0	0	1	0	1	0	0	0	1	1	0	0	2	2	2	3	1
0	0	1	0	1	0	0	0	0	1	0	0	1	3	3	0	0
0	0	2	0	1	1	0	0	0	1	0	0	2	2	1	1	0

Figure 23: Examples of the urgent patient waiting time output. The first two rows show the maximum, minimum and average. Every column is a replication, every row is a patient. The number is the hours of waiting time per patient (or scan).

Appendix F.4: Total makespan output examples.

		Max	34:50:00	Avg	30:06:24												
		Min	23:50:00														
CMAXLEPT																	
Hours																	
34:50:00	33:35:00	34:30:00	27:05:00	34:10:00	33:05:00	28:40:00	28:50:00	27:05:00	31:55:00	28:35:00	28:55:00	23:50:00	25:30:00	29:55:00	32:15:00	28:15:00	32:05:00
		Max	41:05:00	Avg	35:31:48												
		Min	27:30:00														
CMAXSEPT																	
Hours																	
41:05:00	36:00:00	39:05:00	32:40:00	40:50:00	38:50:00	34:05:00	34:30:00	31:05:00	36:25:00	37:50:00	35:15:00	27:30:00	31:20:00	36:00:00	37:15:00	33:00:00	37:15:00

Figure 24: Examples of the total makespan output per heuristic. The first two rows of each heuristic show the maximum, minimum and average of the simulation run. The row beneath the heuristic is filled with every total makespan per replication for calculation purposes.