SIMULATION OPTIMIZATION OF OPERATING ROOM SCHEDULES FOR ELECTIVE ORTHOPAEDIC SURGERIES

DARIA MALTSEVA

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

MASTER OF ARTS

GRADUATE PROGRAM IN
INFORMATION SYSTEMS AND TECHNOLOGY
YORK UNIVERSITY
TORONTO, ONTARIO
AUGUST 2024

Abstract

The aim of this thesis was to solve the problem of scheduling elective surgeries in a multiple operating room setting with the goals of minimizing the amount of overtime incurred. While surgical durations cannot always be perfectly estimated and vary by procedure and surgeon, we propose an approach that relies on leveraging the stochastic nature of surgical durations to simulate each operating day and understand the probability of incurring overtime under a certain schedule of surgeries. The heuristic optimization component of our approach strategically re-schedules surgeries. Through experimentation with three optimization techniques, two showed promising results being able to reduce the total number of overtime surgeries by 12-15%, equivalent to approximately 1h of total monthly overtime. This approach serves as a tool for improving schedules and can be used for supporting decision making at any hospital dealing with elective surgeries. Our contribution involves introducing the simulation optimization model and describing the data-driven approach to analyzing the scheduling problem.

Acknowledgements

It's a pleasure to thank those who made this thesis possible. First and foremost, I'm grateful to my supervisor, Dr. Stephen Chen, for his continuous guidance, ideas, wisdom, and invaluable advice navigating the research world. His cheerful in-person meetings have made this journey truly enjoyable.

I would like to thank the Sunnybrook team for providing the necessary data and extend my gratitude to Cari, Azaad and Johnathan for all their ideas, support, and domain expertise. Their collaboration and insights have been instrumental to this work.

My sincere thanks also goes to my family for their unwavering on-call support from thousands of kilometres away, my peers and friends for the good times we shared and, lastly, myself for not giving up, staying curious, and being patient with the process.

Table of Contents

List of Tables

List of Figures

1. Introduction

The estimated total health spending in Canada for 2023 reached $344 billion, marking a notable increase from $213 billion in 2013 [1], with operating rooms accounting for a significant amount of the total expenditures due to expensive equipment and staffing. While COVID-19 had a noticeable impact on the number of surgeries being performed (approximately 13% fewer surgeries between 2020 and 2022 compared to the pre-pandemic era), the delayed demand for hospital services is expected to drive a rebound in spending growth in the upcoming years [1]. As hospitals grapple with managing rising costs and backlogs, patients face prolonged waiting times and delayed access to surgical procedures.

Prolonged waiting times for total knee and hip arthroplasty (TKA and THA, respectively) lead to a progression of the disease, a decline of the post-surgery outcomes, and a diminished quality of life due to debilitating pain and mobility issues [2, 3, 4]. The provincial and territorial ministries of health developed a benchmark waiting time for total hip or knee joint replacements, that they be carried out within 6 months [5]. Given the rising costs and prolonged wait times for surgical procedures, efficient scheduling is paramount. One way to achieve this would be to minimize minutes of overtime for the said surgeries while maximizing the utilization of operating rooms.

Previous researchers have tackled the problem from different perspectives and areas. Operating Room (OR) planning and scheduling has been receiving attention from diverse research fields, including computer science [6, 7] and operations management [8, 9]. Exact algorithms [10, 11] and approximate or heuristic algorithms [8, 12, 13, 14] have been widely used in surgical scheduling. Fewer researchers have focused on using simulations in conjunction with optimization in order to solve the scheduling problem. While many works rely on discrete-event simulations (DES) and Monte Carlo simulations [15] to assess the robustness of scheduling algorithms under uncertainty, fewer works use simulations as a basis for understanding the scheduling scene and using that information for optimizing surgical schedules. This thesis develops a simulation optimization framework that utilizes insights from the simulation component in order to drive the optimization of surgical schedules.

The exploration of optimization of surgical schedules is of interest to the academic community as much as it is to the real world, having a need for advanced practical solutions. Practical projects enable us to ultimately translate the benefits from theory to practise [16], however, we've seen a limited number of such works. Mainly, this is attributed to the strict policies surrounding patient privacy that hinder the availability of open-source data relating to hospital scheduling initiatives. Consequently, a significant portion of research relies on generated data or a combination of de-identified real-world and synthetic data.

Nevertheless, even when real data is used, surgical schedule optimization algorithms often encounter obstacles that prevent them from being used in the real-world. For instance, solutions developed using commercial tools can be financially burdensome for hospitals, while customized solutions and algorithms present challenges related to adoption and learning curves. Introducing new setups and processes is a complicated process, especially in the critical domain of healthcare.

The work presented in this thesis has been done in collaboration with the Holland Centre, a stand-alone hospital in Toronto focused on elective orthopaedic surgical care, which performs the highest volume of hip and knee arthroplasty in Canada. This thesis focuses on solving the problem of efficient scheduling of elective surgeries by applying a simulation optimization framework along with using stochastic surgical durations. The approach effectively addresses inefficiencies by evaluating the initially constructed schedule and rescheduling surgeries that are most likely to result in overtime. The novelty of the approach is attributed to the load balancing heuristic that aims to distribute planned surgeries within a short scheduling horizon, minimizing the number of surgeries ending in overtime and maximizing OR utilization throughout the day.

The remainder of the thesis is organized as follows: Chapter 2 introduces some background information about surgery planning and scheduling. Chapter 3 dives into a review of existing literature on heuristic optimization techniques, simulation optimization solutions and stochastic surgical estimations. The section also discusses the limitations of existing work and how our approach is different. Chapter 4 presents the findings from the problem analysis and contains a detailed overview of the methodology. Chapter 5 and 6 present experiments with surgical duration distributions and turnover time predictions, respectively. Chapter 7 describes the discrete-event simulation model. Chapter 8 describes the optimization models and presents the results of the simulation optimization experiments. Finally, Chapter 9 concludes the research and discusses the topics that could be expanded in the future works.

2. Background

2.1 Planning and scheduling surgeries

An operating room (OR) refers to a medical facility designed and equipped to perform surgeries. The process of planning and scheduling surgeries within the ORs is typically divided into several decision levels. Depending on the organizational structure of the hospital these decision levels may vary but generally each hospital distinguishes between long term (strategic level), medium term (tactical level), and short term (operational level) planning and scheduling. A quick overview of the levels is shown in Figure 2.1.

The long-term scheduling problem is referred to as the "Case Mix Problem (CMP)" in which the goal is to allocate a certain amount of OR time to a surgical specialty in order to optimize the profit and costs over a long time [17, 18] or to meet expected long-term patient demand [14, 19]. The decisions made at this level include determining the number and specialties of surgeries to be planned, the amount of resources required, and allocating the amount of operating room time to the various specialties. The time-frame of these decisions

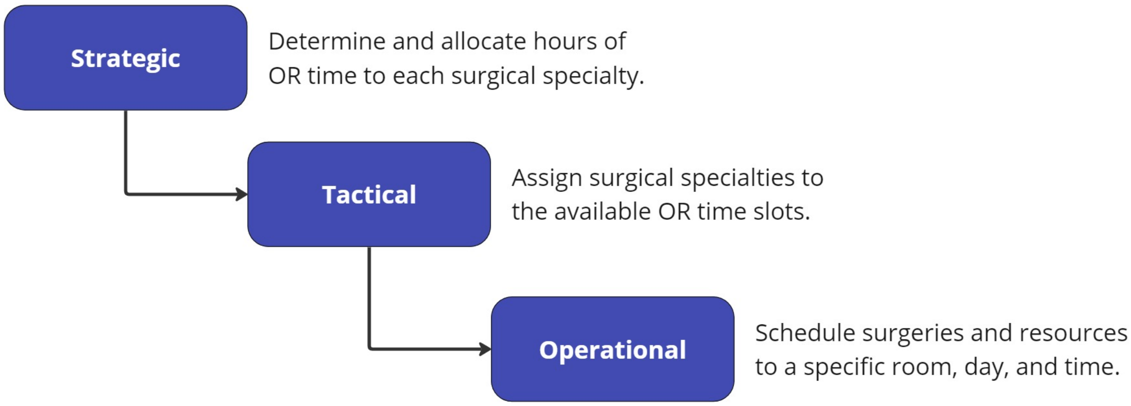


Figure 2.1: Overview of scheduling decision levels.

could span from several months to one year or longer [20].

The medium-term scheduling problem is referred to as the "Master Surgery Scheduling (MSS)" problem in which the goal is to assign surgical specialties to the OR time slots in order to optimize the levels of resource utilization [6, 21, 22]. MSS is known as a cyclic schedule and it is usually monthly or quarterly. The decisions made at this level include determining the distribution of the workload which is restricted by the capacity and demand constraints determined at the strategic level. The tactical level provides guidelines that facilitate decisions at the short-term operational level.

The short-term scheduling problem is referred to as the "Operational Level" scheduling or the "Surgical Case Scheduling (SCS)" problem in which the goal is to assign resources (surgeons and nurses) and patients to a specific day, room, and time. The problem is typically decomposed into two steps, namely advance scheduling and allocation scheduling. Advance scheduling is concerned with assigning a definite date for each operation (and room, in some cases), while allocation scheduling is concerned with determining the exact start time of the operations and the allocation of the OR resources.

At the operational level, it's crucial to accurately estimate surgical durations and plan an efficient OR usage, without incurring overtime or wasting undertime. This thesis will focus on advance scheduling but from a rescheduling perspective. Given the unpredictability of surgical durations, OR efficiency is often assessed after the day has ended. However, the proposed simulation optimization method allows for the evaluation of OR utilization before surgeries take place. The approach involves assessing an initial surgery schedule and recommending rescheduling certain procedures to different days to prevent over- or under-utilization of resources based on a probabilistic analysis of OR usage.

2.2 Stages of the surgery

The overall surgery process involves several activities before, during, and after the actual surgical procedure [23]. These include pre-operative, intra-operative, and post-operative stages illustrated in Figure 2.2. Pre-operative care begins with the patient and surgeon deciding to have the surgerical procedure. This might include a visit to a preoperative clinic for exams and tests. Preparation before the surgery day is also common. On the surgery day, the patient arrives at the hospital at a scheduled time. Various administrative tasks may be needed, and the patient might undergo a brief physical exam or medical tests (e.g., a blood

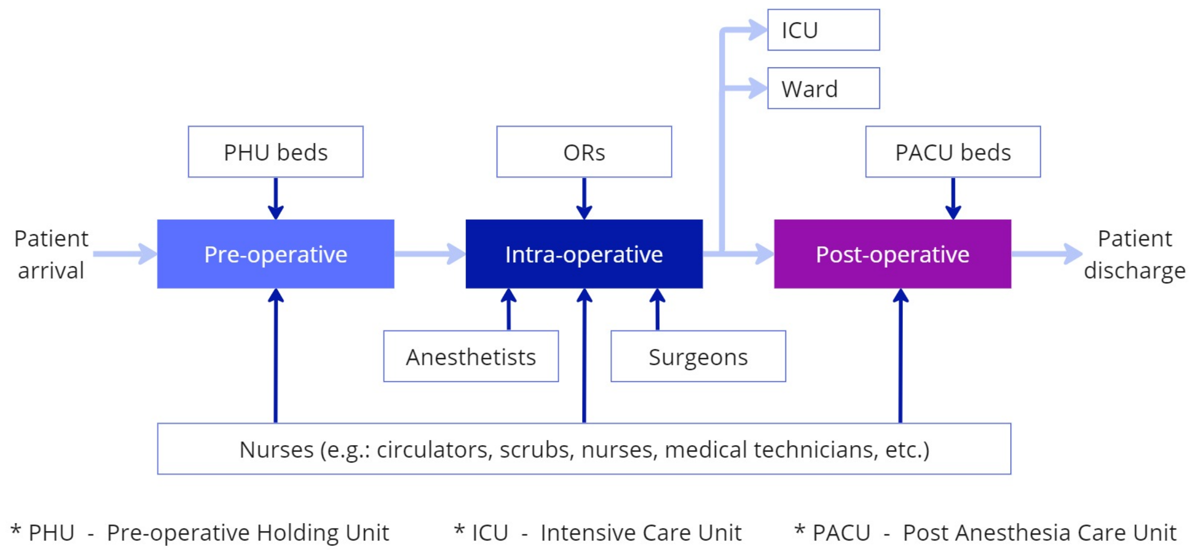


Figure 2.2: Stages of surgery process.

test) before surgery.

Intra-operative care refers to activities that happen in the OR. This includes the patient receiving anaesthesia, undergoing surgical activities, and waking up from the anaesthesia. The patient triage post-operative stage starts when the patient is admitted to a recovery area, either a Post Anaesthesia Care Unit (PACU) or Intensive Care Unit (ICU), depending on the surgery's nature. The whole surgical process ends when the patient has fully recovered, and no further follow-up with the surgeon is needed. This could take weeks, months, or even years. In this thesis, we focus on considerations and scheduling decisions made within the intra-operative stage – our dataset does not include the availability of PHU and PACU beds, so these constraints are beyond the scope of our current problem domain.

2.3 Evaluating schedule performance

Different evaluation metrics can be used to evaluate the performance of OR schedules. The choice of the measures reflects the needs of the stakeholders and the most important objectives of the problem. For instance, patients are predominantly worried about their surgery being cancelled and long waiting times. Whereas hospital staff are more concerned about overtime, shift assignments (both in terms of time and colleagues they are working with during the shift), and block time allocation [16]. The two most common measures that have been used the most in the literature between 2015 and 2020 have been patient waiting time and overtime. These two are also commonly selected with other popular metrics such as rate of OR utilization, financial measures, throughput, and others presented in Table 2.1 [16, 24, 25].

Table 2.1: Common metrics used to evaluate performance of operating rooms.

Metric	**Description**
First case start time	Percentage of first cases of the day that start on time.
OR utilization	Percentage of OR time used against that which was budgeted.
OR turnover time	Average time elapsed between surgeries in one operating room.
Case duration accuracy	Percentage of OR cases with durations that are accurately estimated.
Excess staffing costs	Staffing costs associated with underused and overused OR time.
Off hours / overtime surgery	Volume, percentage, or duration of surgeries performed outside of scheduled OR time.
Procedure cancellation rate	Percentage of procedures cancelled on the day of the surgery.
Percentage of unplanned closure	Percentage of OR time lost due to unplanned closures (lack of human or hospital resources, environmental factors, etc).

Focusing only on one of the performance measures can impact the performance of the others. For instance, focusing only on patient-centred metrics can lead to more satisfactory patient outcomes but may negatively impact hospital revenue and lead to increased OR idle times. Conversely, focusing only on hospital management metrics may increase profitability at the expense of increased patient wait times, cancellations, and/or dissatisfied OR staff. Researchers often turn to multi-objective modelling to balance the competing priorities [16]. In recent years, operating room utilization, patient wait times, and overtime have been the most popular measures [20].

This work will predominantly focus on minimizing overtime. By minimizing overtime, we aim to have a positive impact on OR utilization and staff workload. As ORs experience less overtime and undertime, more surgeries can be performed, leading to reduced patient wait times.

3. Literature Review

Researchers started to address operating room planning and scheduling problems starting in the late 1970s [26]. Many approaches and techniques have been proposed since then. Our research is focused on utilizing a simulation optimization approach which addresses uncertainty in surgical durations. Therefore, we dedicate a major portion of this chapter to review the articles about heuristic optimization, simulation optimization, surgical durations, and rescheduling.

3.1 Heuristic optimization

Heuristics are a set of procedures prioritizing efficiency and speed over guaranteed optimality. Heuristic algorithms typically fall into six broad categories: heuristics based on exact methods, constructive heuristics, improvement heuristics, metaheuristics, linear programming-based heuristics, and dispatching-rule based heuristics [15].

Heuristics based on exact methods, as the name suggests, are inspired by exact methods. Exact methods, such as Integer Programming (IP), are guaranteed to identify and verify an optimal solution but by doing so they can become computationally expensive. In practice, it is common to trade the guarantee of optimality for efficiency [27]. As such, hybrid solutions combine heuristic algorithms with IP techniques like linear programming (LP), mixed integer programming (MIP), dynamic programming, and others. Hybrid solutions may be "collaborative" in nature, in which information is shared between algorithms but they are ran separately. Alternatively, the solutions may be "integrative" in nature, where one technique is an embedded component of the other. For instance, Jung and colleagues [10] proposed a model that combines MIP and heuristic scheduling to create daily schedules, offering precise solutions while considering the dynamic nature of surgical operations. Similarly, Astaraky and Patrick [28] propose an approximate dynamic programming method to minimize waiting time and resource congestion, highlighting the adaptability of exact methods to real-world scheduling challenges.

Constructive heuristics involve methods that build feasible solutions from scratch, with the primary goal of constructing a solution by iteratively adding elements one at a time. A two-phase iterated constructive algorithm proposed by Molina-Pariente and colleagues [29] demonstrates an iterative construction and improvement of a solution. The first phase is concerned with creating a fast and feasible surgery schedule by constructing the schedule based on specific criteria. The second phase is focused on improving the solution by iteratively invoking the construction surgery schedule step, randomly selecting surgeries from the waiting list, assigning new assistant surgeons and re-calculating the weighted objective function. The iterative process continues until a CPU time limit, dependent on the problem size, is reached. Spratt and Kozan [30] considered a real-time reactive surgical case sequencing problem with the objective of maximizing total OR utilization. They proposed two constructive heuristics: modified block scheduling and modified open scheduling. In their modified block scheduling approach, elective patients are initially assigned to surgeon blocks based on the recommended surgical due dates. Then, non-elective patients are added to these blocks at the earliest possible time. This process repeats for each specialty and patient throughout the day. In contrast, their modified open scheduling method evaluates all possible surgeon-patient-OR assignments to determine the earliest feasible surgery start time, without strict adherence to surgeon-patient assignment.

Improvement heuristics involve methods that start with an initial solution and aim to iteratively refine and improve the solution by making local changes. Lamiri and colleagues [31] proposed sequential improvement, local optimization, and pair-wise switching methods to iteratively refine surgical schedules based on OR utilization, emphasizing the iterative refinement process. Their pair-wise switching heuristic always outperformed local optimization, yet local optimization was always better than the sequential improvement heuristic. Similarly, Marques and colleagues [32] addressed the problem of OR utilization by applying a genetic algorithm with constructive and improvement heuristics. Their iterative approach addressed not only the surgical suite occupation rate but also the priority of surgeries to accomplish their goals.

Metaheuristics are high-level algorithms that are applicable to a wide variety of problems, often chosen for combinatorial optimization problems that require searching over a large set of feasible solutions. For instance, Xiang and colleagues [33] introduced Ant Colony Optimization (ACO) to address operating room surgery scheduling challenges. By modifying the original ACO and introducing a two-level hierarchical ant graph representing surgery sequencing and resource allocation, the algorithm efficiently determines surgery start times while considering various constraints. Comparative experiments demonstrate ACO's superior performance in minimizing makespan, overtime, and working time variation compared to a discrete-event simulation model of the original schedule. Similarly, Farsi and colleagues [34] tackle daily scheduling issues by integrating constraint programming (CP) with evolutionary algorithms like NSGA-II and the multi-objective dragonfly algorithm with the objective of reducing makespan while enhancing both patient and surgical teams' satisfaction. While the CP model suits smaller hospitals with less than 150 surgeries per week, their metaheuristic approach proves more effective for larger-scale and multi-objective problems.

Dispatching-rule based heuristics are those that rely on predefined rules, such as first-come-first-served, longest processing time, or a specified priority. Wang and colleagues [35] propose an adaptive composite dispatching method, combining popular dispatching rules to minimize makespan. Their combination of Longest Processing Time first (LPT), Least Flexible Job first (LFJ) and Largest Resource Workload first (LRW) was shown to significantly shorten the makespan and reduce the overtime work when testing the results through a case study using real data. Similarly, Pham and Klinkert [36] extend traditional scheduling rules by building on top of a classical Job Shop scheduling problem (e.g. the multi-mode blocking job shop (MMBJS)) for solving the surgical case scheduling problem. Their mixed integer linear programming formulation of the MMBJS considers the availability and need of different resources depending on the surgical stage (pre-operative, intra-operative, and post-operative) and manages emergency cases, as well as add-on elective cases, adaptively and efficiently.

The widespread application of heuristics in surgical scheduling underscores its acceptance as a viable problem-solving tool. However, due to the nature of the surgery scheduling field, the majority of evaluations of these techniques have predominantly been numerical or theoretical, with limited testing on real-world data. Strict patient privacy policies limit the availability of open-source hospital data causing researchers to rely on generated or a combination of de-identified real-world and synthetic data. Our objective is to contribute to the field of heuristic algorithms by introducing novel improvement heuristic algorithms. We aim to leverage the potential of heuristics to optimize schedules, considering their inherent advantages in terms of resource efficiency and flexibility. We will validate the results not only in a simulated environment but also through a comparative analysis with real hospital data.

3.2 Simulation optimization

Simulation and optimization have traditionally been considered as two different approaches. On one hand, simulation is the process of designing a model of a real system and conducting experiments with it. The goal is to either understand the behaviour of the system or to evaluate various strategies. On the other hand, optimization is the process of finding a solution that conforms to predefined constraints and reaches the ultimate goal or objective. However, increasing computational power has been enabling the combination of both of these approaches together [16, 37]. Xiao and colleagues [38] developed a simulation optimization approach to find heuristic solutions for the surgery scheduling problem and have devised several scheduling policies useful for hospital planning teams. They developed a two-step approach where initial schedules were created by a simulation optimization approach, and then numerically evaluated by different sequencing and allocation policies using a discrete-event simulation model. Bovim and colleagues [21] also developed a simulation optimization model, however, in their case the simulation model was used as input for the optimization model. The proposed method was run repeatedly, creating a feedback loop, until a certain stopping criterion has been met.

Simulation optimization is a very wide term used in research since there are different ways simulation can interact with optimization and vice versa. Thus, simulation optimization is a field of its own that can be classified into several dimensions. The taxonomy proposed by Figueira and Almada-Lobo [37] classifies simulation-optimization methods into four key dimensions: Simulation Purpose, Hierarchical Structure, Search Method, and Search Scheme employed.

Purpose and structure both relate to the interaction between simulation and optimization.

When combining the approaches, the purpose of the simulation can be different. It may be used to evaluate solutions, to generate solutions, or to combine both of those approaches. The extent and nature of interaction between simulation and optimization components depends on the structure. Both modules could run sequentially (Sequential Simulation-Optimization) or could alternate in each iteration (Alternate Simulation-Optimization). Moreover, simulation could be simply a part of an optimization procedure (Optimization with Simulation-based Iterations), or alternatively optimization could be a part of a simulation process (Simulation with Optimization-based Iterations). Liang and colleagues [39] employed a sequential simulation–optimization to maximize patient throughput and minimize patient waiting time. The purpose of their simulation model was to evaluate the scheduling performance of three simple scheduling rules, namely shortest processing time (SPT), critical ratio (CR), and first-come first-served (FCFS) rules. The structure relied on simulation to obtain evaluation results which would be passed to a response surface methodology (RSM) to determine an optimal weights configuration of the simple rules and to a Tabu search optimizer to find an optimal combined scheduling policy, finally selecting the best policy.

Search method and scheme are more concerned with search algorithm design. The search method specifies the type of optimization method that could be employed: an exact method or any type of heuristic methods (which can also be categorized into several types: Derivative-Based Heuristic, Other Continuous Heuristic, Discrete Heuristic). While the search scheme determines how the optimization algorithm navigates through the solution and the probability space to find the best possible solution. Landa and colleagues [40] proposed a hybrid optimization approach that uses heuristic techniques combined with a Monte Carlo simulation to attempt to achieve an acceptable OR utilization rate while limiting the negative effects of surgery cancellations and delays. Their two-fold optimization technique started off with a Local Search for Feasibility module, followed by a Tabu Search for Improvement module. In a paper by Lamiri and colleagues [41], instead of a heuristic search, the authors combined Monte Carlo simulation with a Mixed Integer Programming (an exact method), aiming to minimize the expected overtime costs as well as costs related to elective cases (waiting time costs, hospitalization costs, etc).

In surgical scheduling, Monte Carlo and discrete-event simulation have both been utilized, but they differ in key design aspects. Monte Carlo techniques are suitable for continuous systems with uncertain parameters and have been predominantly applied in industrial engineering, physical processes, economics, and finance areas, where stochastic sampling drives simulation [42]. In surgical scheduling, Monte Carlo techniques have been useful for scenario reduction and modelling emergency patient arrivals. Lamiri and colleagues [41] used Monte Carlo methods to simulate the arrival of emergency patients to the hospital, similar to Antogini and colleagues [43] who also simulated non-elective arrivals with random patient classes, surgery times, and wait times using Monte Carlo methods. Pulido and colleagues [44], on the other hand, used Monte Carlo simulation to emulate 100 scenarios obtaining an optimal value, thus reducing the number of scenarios that will need to be used in the solution and decreasing solution time.

Discrete-event simulations, on the other hand, are suitable for systems with distinct events and state changes and have been applied in manufacturing processes and queuing systems, where the system is driven by discrete events triggering state transitions. Discrete-event simulations (DES) have been popular within surgery scheduling circles. Ma et al. [17] built a DES model through Arena (a simulation software) with random elective patient arrivals and random patient requirements achieved by adopting probability distribution functions. Arriving patients are put into a waiting queue and then allocated to a proper surgical centre to be operated on by a specific surgeon. After surgery, patients are moved to a proper recovery ward. The simulation modelling of the hospital is portrayed from the perspective of the patient flow. McRae and colleagues [18] modelled their discrete-event simulation in Java 8 with stochastic elective and non-elective patient arrivals, surgery durations, and lengths of stay in the intensive care unit (ICU).

3.3 Surgical durations

Surgical durations are one of the major categories of uncertainties that hinder the accuracy of scheduling practices. The uncertainty of surgery duration is mainly attributed to the inability to accurately predict the effect of patient conditions and the effect of surgeon experience on the duration of the surgery [15]. Furthermore, the duration depends not only on the specialty, such as orthopaedic, cardiac, or neurological but also the specific procedure being performed, such as interventions on the hip, on the knee, or other bodily parts, and anaesthetic being used. Thus, the uncertainty of actual surgery duration contributes to the challenge of accurately forecasting it.

While some researchers focused on solving this problem deterministically, others turned to stochastic modelling. Deterministic approaches, such as machine learning techniques, have been highly popular in the surgical field. Techniques such as Random Forest Regression, Support Vector Machine, Linear Regression, Artificial Neural Networks (ANNs), and Naive Bayes have been applied for predicting the duration of surgeries based on preoperative factors [45, 46]. Different machine learning models continue being experimented with in order to better incorporate the non-linear relationships between the variety of factors that affect surgical durations.

Nonetheless, while deterministic approaches aim to accurately predict durations, stochastic approaches aim to estimate surgical durations and incorporate an amount of variability that comes with them. Thus, stochastic approaches have been more popular within the simulation optimization field where variability and uncertainty are crucial components in planning and scheduling. A widely adopted technique for such purposes has been to draw surgical durations from probability distribution functions (PDFs).

In summary, a PDF is a mathematical function that gives the probabilities of occurrence of different possible outcomes for an experiment. Surgical durations are typically modelled as continuous variables, with the probability of a certain outcome depending on the probability of its specified interval. The majority of researchers have modelled their surgical durations as one type of a PDF, such as a log-normal distribution function [9] or an exponential distribution function [47] which considers the property that typical surgical duration distributions have long tails. However, this approach lacks the consideration that certain surgical procedures can fit different PDFs. Only a minority of works incorporate this consideration and fit their surgical durations to several PDFs, depending on which function better fits the specific procedure [48, 49, 50].

3.4 Surgery rescheduling

Surgery rescheduling is a crucial aspect of operations aimed at optimizing the use of operating rooms (ORs) and minimizing costs, based on either additional knowledge or disruptions that arise after the creation of an initial surgical schedule. Rescheduling stems from a desire to reduce uncertainty and can arise at the tactical or at the operational level. At the tactical level, there might be changes to the required capacity or demand for a specific discipline or surgeon. At the operational level, there might be stochastic arrivals of urgent or emergency cases, case cancellations, or deviations to scheduled case durations [51]. The uncertainty can be managed from a proactive or reactive approach. In proactive approaches, managers try to anticipate disruptions and plan to minimize their impact, whereas in reactive approaches, the surgical schedule is changed only if the disruptions occur.

The majority of literature focuses on the on-line reactive approaches, re-scheduling elective cases upon the arrival of emergency cases or prolonged surgery durations [52]. Eshghali and colleagues [53] introduced a three-phase reactive model for surgery planning with emergency arrivals. In the first phase, surgeries are scheduled a week in advance. The second phase determines the exact timing and sequence of the surgeries. In the third phase, if an emergency case arises during the day, the remaining subset of scheduled surgeries for the day are rescheduled. Their model ensures that the rescheduling is done within a reasonable time limit and that the original surgery day for patients remains unchanged, which helps reduce patient dissatisfaction and anxiety. The model also accounts for the use of multiple operating rooms and includes constraints related to recovery bed availability. Allen and colleagues [54] proposed a simulation model which reactively reschedules surgeries in an operating room, if a procedure runs late beyond a set criterion. The study examined the start and end time offsets, the number of rescheduling events, and their impact on schedules. Results showed that increasing the set criterion reduces the number of rescheduling events, with an optimal criterion of 10-minutes for their particular hospital.

A proactive rescheduling approach was proposed by Ballestin and colleagues [55] which consisted of two phases. In the first phase, a tentative schedule is built 2 weeks in advance to minimize the the percentage of tardy patients (those who are scheduled after their estimated due date). In the second phase, a final schedule is made a few days before the planning period as a consequence of rescheduling certain surgeries based on the changes in the available information (new arrival of patients or unavailability of scheduled patients, surgeons, equipment, etc.).

This thesis focuses on a proactive rescheduling approach, where we begin with an initial surgical schedule prepared by the hospital staff and run it through the simulation model to obtain a probabilistic evaluation of the schedule efficiency. Once the overtime and undertime are quantified, the proposed optimization model reschedules surgeries to balance the schedule and reduce the amount of overtime incurred.

3.5 Contribution of the thesis

While the field of surgical scheduling is not new and has been of interest to researchers for some time now, the field of simulation optimization remains relatively understudied despite being highly promising for this area [16, 37]. A majority of simulation optimization works in surgery optimization have focused on using simulations for performance evaluations, and less work integrates both modules to depend on each other. The approach presented in the thesis relies on combining simulation findings with an algorithm that reshuffles surgeries to minimize the minutes of overtime incurred. The use of stochastic durations incorporates a level of uncertainty that's useful for estimating variability of the real world and utilizing it for creating an efficient schedule.

In addition, there haven't been many works relying on real-word data, making it harder to translate the ultimate benefit of theory to practice. The work in this thesis relies on the data provided by a local hospital and retains the assumptions posed from the real world.

Thus, the second contribution of this thesis includes a data-driven approach to solving an elective patient scheduling problem, making use of a discrete-event simulation model and a heuristic optimization algorithm. Heuristics and simple scheduling rules gathered from this approach could be useful not only within the algorithmic model but also easily applied by hospital administrators for surgical scheduling.

4. Problem Analysis and Formulation

4.1 Problem Description

We consider the problem faced at the the Holland Centre (part of Sunnybrook Health Sciences Centre), a stand-alone hospital in Toronto, performing the highest volume of hip and knee arthroplasty in Canada. Inefficiencies of surgical schedules arise mainly from the inability to accurately estimate surgical durations, as well as from delays occurring due to cleaning and preparation activities between surgeries inside a room. Over and under estimation of durations has a large effect on the over and under utilization of the ORs, which entails financial implications and disruptions to the schedule of the staff. Our goal is to investigate the problem and address the root cause to create more optimal schedules.

This research is split into three phases. First, we analyze the current state of the scheduling practise. Our aim is to understand and quantify the problem of turnover times, overtime and undertime. Second, we build a simulation that models the hospital operations and allows us to deepen our understanding of the relationship between scheduled surgeries and performance metrics. Third, we create an optimization algorithm that addresses the specific problem with the current schedules and proposes an improved schedule to avoid delays.

4.2 Problem Analysis

In the beginning of our problem exploration, the main issue seemed to arise from the delays between surgeries. As such, our problem analysis begins with an exploration of trends and correlations related to turnover times. We identify one major predictor of turnover time delays: concurrently ending surgeries, which impose a a strain on the environmental team who are then called on to clean several rooms simultaneously. Thus, we analyze and quantify the problem of concurrent ends. Finally, we turn our attention to the problem of overtime and undertime, both of which highlight a greater source of inefficiency.

We obtain the data for the analysis from the Hospital's Decision Support over a 10-year period. It comprises of information on each surgery, including procedure descriptions, estimated and actual surgery durations, timestamps for when each patient entered and exited the operating room, and when each procedure started and ended. A full list of features used is presented in Table 4.1. There were several important features in the dataset that weren't included nor collected by the hospital. For instance, there was no record of which room the surgery was performed in or how long the turnover time was between surgeries. Since these features are critical for our analysis, we had to infer them from the data based on certain assumptions. To do that, we first had to transform the data into a proper format.

Table 4.1: List of features extracted and calculated from the hospital dataset.

Feature	Description	Type
Surgery ID	Surgery identifier	Extracted
Patient ID	Patient identifier	Extracted
Date	Date of surgery	Extracted
Surgery Procedure	Procedure code of the surgery	Extracted
Estimated Surgery Duration	Manual duration estimation of the surgery	Extracted
Actual Surgery Duration	Actual duration of the surgery	Extracted
Patient Wheeled Into Room	Time when patient is wheeled into the OR	Extracted
Procedure Start	Time when surgeon begins operating	Extracted
Procedure End	Time when surgeon stops operating	Extracted
Patient Wheeled Out of Room	Time when patient is wheeled out of the OR	Extracted
Turnover Duration	Time between one surgery end and the next surgery start in one room	Calculated
Room	Room number where surgery takes place	Calculated

4.2.1 Transforming and enriching the data

The original dataset was structured the following way: each surgical case occupied one row and each column represented surgical features described in Table 4.1. However, this format was not suitable for our analysis for one main reason - the sequence of surgeries for the day was not obvious from such a format. We needed to have an ability to sort surgeries by their start times to calculate the missing features of Room and Turnover Time. Thus, a transformation of the data into an event log format was needed.

Event logs are structured files containing a chronologically ordered list of the recorded

Surgery ID	Patient In Room Timestamp	Procedure Start Timestamp	Procedure End Timestamp	Patient Out of Room Timestamp	Procedure Code	Surgeon ID
1	8:00 am	8:10 am	9:10am	9:13am	555	00800
2	8:30 am	8:40 am	9:43am	9:55am	333	00700
3	9:15am	9:20am	9:55am	10:05am	444	00600

Surgery ID	Activity	Timestamp	Procedure Code	Surgeon ID
1	Patient In Room	8:00 am	555	00800
1	Procedure Start	8:10am	555	00800
2	Patient In Room	8:30 am	333	00700
2	Procedure Start	8:40 am	333	00700
1	Procedure End	9:10am	555	00800
...				

Figure 4.1: Transposing the original data to event log format.

events. Each event in the log has an ID, an associated timestamp, and an event name. Event logs are a foundation of process mining - a technique used to discover, analyze and improve real processes by extracting knowledge from event logs [56]. To transform our original dataset, mindize software was used [57]. Each row representing one surgical case was split into several rows, representing each separate event related to the surgery such as patient wheeled into the room, procedure start, procedure end, and patient wheeled out of the room. A visual representation of the transformation is shown in Fig. 4.1.

Following the transformation, the dataset was sorted by the timestamp of each event. This allowed us to establish a chronological order of surgeries. This served as a foundation for introducing new key features. Firstly, we enumerated each surgery in the operating theatre from the first to the last surgery per day and called it "Surgery Number Today". Secondly, we introduced features to store information about the assigned room for each surgery and the turnover time between consecutive surgeries in the same room. To calculate the "Room" and "Turnover Time" features we initialized two dictionaries. First, to store surgeon-to-room assignments throughout the day (needed for the "Room" feature) and, second, to track the index of the last surgery in each room (needed for the "Turnover Time" feature). We iterated through each row, identifying the current date. For each date, a list of unique surgeons was compiled and each surgeon was assigned to a room randomly. Surgeries were assigned to rooms throughout the day based on the surgeon performing the operation. Parallel to the assignment of surgeries, turnover times were calculated and assigned retrospectively. To illustrate this, let's say surgery #1 ended in a room at 9:50am and surgery #2 started in that same room at 10:22am. The turnover time of 32 minutes was calculated and assigned to the row corresponding to surgery #1 in the dataframe. Last surgeries were assigned a turnover time of 0 minutes, since the next surgery in that room only occurred the following morning.

Subsequently, we introduced several features relating to surgeries ending concurrently with one another. Concurrently ending surgeries are those that ended within 5 or 10 minutes of another surgery in a different room. To calculate the feature, we introduced two boolean features, namely 'concurrent end 5 mins' and 'concurrent end 10 mins'. The features denoted whether a surgery ended concurrently within 5 or 10 minutes of one another. Additionally, two numerical features, 'concurrent end number 5 mins' and 'concurrent end number 10 mins', were introduced to indicate the number of surgeries that ended concurrently within the 5 or 10 minutes interval. The calculation function operated in two steps. First, it traversed through each row, noting the end time of the current row surgery and next row surgery. If the next surgery ended within 5 or 10 minutes of the current surgery, then the corresponding feature was marked as '1' (concurrent end happened); otherwise, it was marked as '0' (concurrent end did not happen). Second, the function revisited each row, focusing solely on surgeries that ended concurrently. For each identified concurrent end, it determined the number of neighbouring surgeries that ended within 5 and 10 minutes of the current row surgery. The iterations continued until no neighbours within a 10-minute range were found, marking the completion of the concurrent ending surgery search.

4.2.2 Turnover times

Turnover time is the average time that elapses between one surgery end for a surgeon and the next surgery start for that same surgeon. At our hospital, the average turnover time between 2012 and 2022 was 25 minutes with a standard deviation of 25 minutes. The large standard deviation has been attributed to the heavily skewed distribution of turnover times, as well as misleading cases that were included into this calculation.

It's important to note for turnover times is that it is a calculated feature, which led us to find some instances of misleading values. Firstly, we identified that sometimes surgeons performed two surgeries in one room or two surgeries in different rooms. As such, the calculation of turnover times for such cases resulted in negative values. Typically, such cases happened at the end of the day, where surgeons had to start their next surgery concurrently to finish their day on time. In total, there were 124 (less than 1%) instances of negative turnover time values. Since these cases are rare, we have completely removed them from the dataset. Secondly, there were instances where turnover time had no value. This occurred for surgeries that happened last in a room for the day. Since the next surgery in that same room doesn't happen until the next day, there's no way for us to estimate how long the cleaning for the room took place. To avoid missing values and to indicate last surgeries in rooms for the day, we assigned a value of 0 minutes to such cases. After removing rare cases and re-calculating the turnover times without last surgery cases, it was found that the average turnover time between 2012 and 2022 was 37.34 minutes with a standard deviation of 21.06 minutes.

To understand turnover time delays more, we looked at cases that incurred very long turnover times, defined as cases with turnover times more than 58.40 minutes (average turnover time plus one standard deviation). In our analysis long turnover times were identified in 2.88% of cases. The mean duration of long turnover times was 133.97 minutes with standard deviation of 60.47 minutes. Typically, compared to all cases, long turnover cases occurred for surgeries that ended between 9am and 12pm, as seen in Fig. 4.2. This is rationalized by the fact that the first half of the day can allow more delays since there's time in the second half to catch up on those inefficiencies.

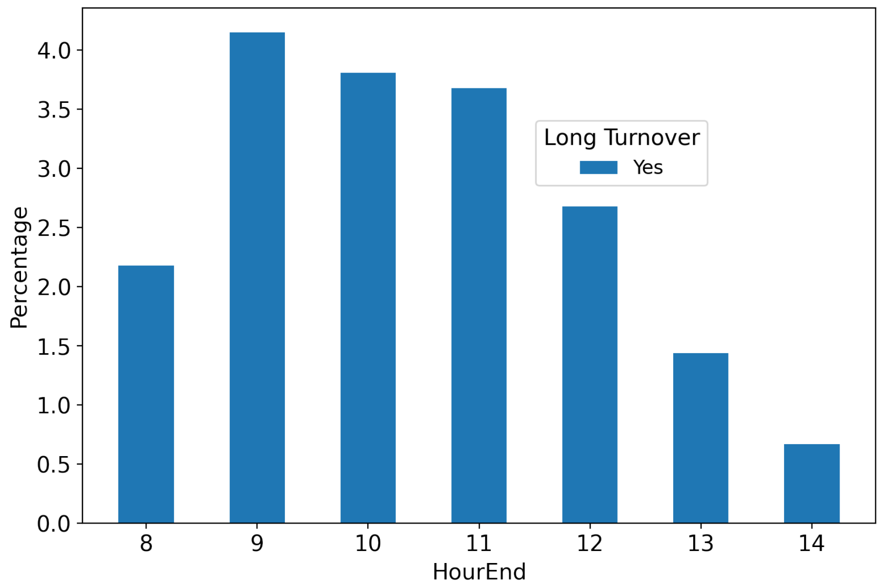


Figure 4.2: Long turnover cases by the hour when surgery ended.

4.2.3 Concurrently ending cases

Concurrently ending cases are those surgeries that end within 5 to 10 minutes of each other in the same operating theatre but different ORs. These cases have an impact on the length of turnover time because they impose a strain on the environmental services team, who are called on to clean several rooms simultaneously. Between 2012 and 2022, there were 2,499 (16.37%) cases of surgeries in different ORs ending within 5 minutes and 4,587 (30.03%) cases ending within 10 minutes of each other. Typically, concurrently ending cases happened between 9am and 10am, as seen in Fig. 4.3. Surgeries tend to end concurrently predominantly in the morning due to most surgeries having the same time in the morning (around 7:30 am - 8:00 am) and having similar surgical durations.

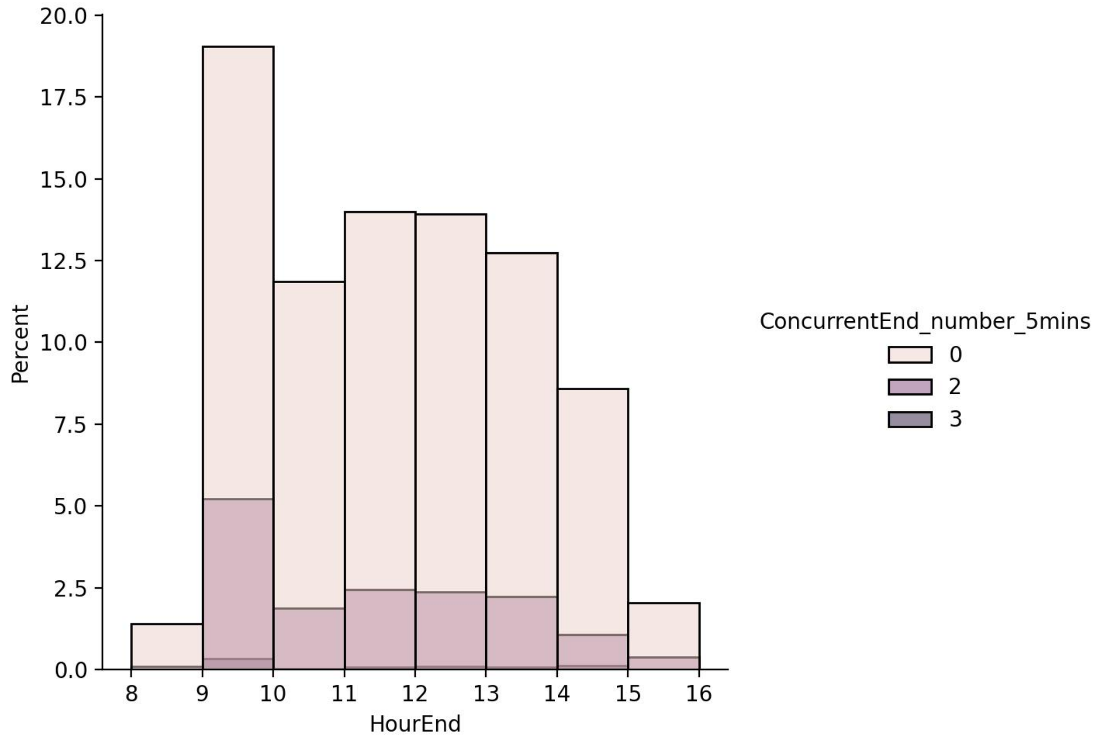
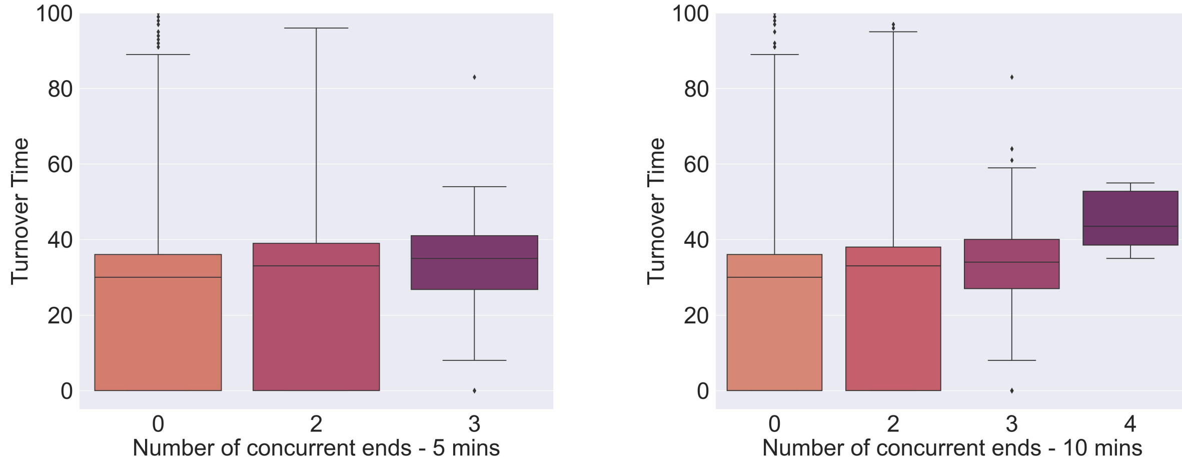


Figure 4.3: Percent of concurrent ends by the hour when surgery ended.

We analyzed the effect of concurrently ending surgeries on turnover time. Box plots shown in Fig. 4.4 show a visible trend where turnover time increases as more surgeries end concurrently with one another. A linear regression analysis was run to examine significance of the observed trend. It was found that surgeries ending within 5 and 10 minutes of each other are both significantly correlated with increased turnover time (p-value <0.01) for all surgeries performed between 2012 and 2022. However, for the more recent data (i.e.: surgeries performed between 2020 and 2022) only surgeries ending within 5 minutes of each other were significantly correlated with longer turnover time (p-value <0.01). This highlights a change in hospital operations, where surgeries ending within 10 minutes of each other don't pose the



(a) Concurrently ending cases within 5 minutes. (b) Concurrently ending cases within 10 minutes.

Figure 4.4: The impact of the number of concurrent ends on turnover time.

same level of strain on the environmental team as 5 minutes cases. In addition, this suggests a trend towards more efficient hospital operations, where concurrently ending surgeries have less impact on turnover time delays.

4.2.4 Overtime

Overtime surgeries are those that finish the procedure past the scheduled room end time. The hospital we worked with scheduled each of their ORs to end at either 3:30pm, referred to as a short room; or at 5pm, referred to as a long room. The OR end times varied daily. In our analysis we look at overtime from January to December 2021, as that is the only period with the room end time label.

In 2021, 230 cases ended in overtime, representing 11.31% of all cases. In total, it resulted in 156.33 hours of overtime for the whole year. On average, each month incurred 13.03 hours of overtime. The summer months of June, July and August incurred 85.51 hours of overtime alone (54.70% of total annual overtime).

When comparing actual ends times to the scheduled end times, we found that the majority of overtime has been incurred by rooms scheduled to end early, at 3:30pm. As seen in Fig. 4.5, rooms that were supposed to end at 3:30pm overrun their scheduled end time by 0.5 - 2.5 hours, sometimes ending even past the long room end time of 5pm. The long rooms also tended to overrun their scheduled ends but never by more than 1.5 hours.

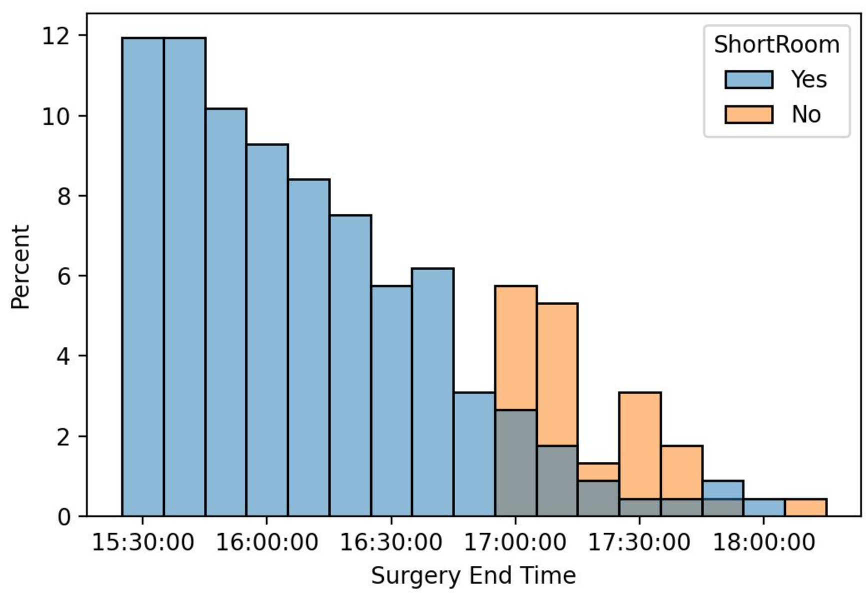


Figure 4.5: Overtime incurred by short (3:30pm) versus long (5pm) rooms.

Typically, for each analyzed month, the amount of overtime incurred was less than the amount of undertime incurred. Table. 4.2 shows the daily average of overtime and undertime during the month, while Table. 4.3 shows the total number of overtime and undertime hours incurred during the month. Fig. 4.6 shows the general trend of undertime exceeding overtime in each month. These findings suggests an opportunity for optimization, where surgeries that end up in overtime are re-scheduled to days when the OR is underutilized.

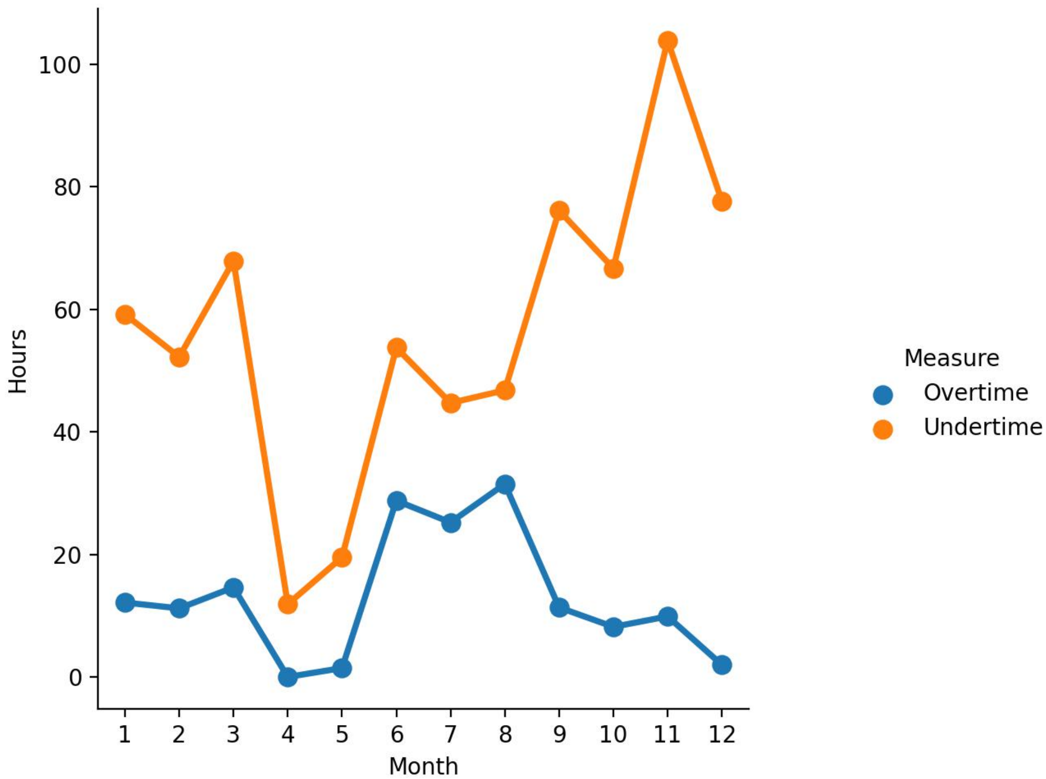


Figure 4.6: Total hours of overtime compared to undertime per month.

Table 4.2: Summary of average overtime and undertime incurred in 2021 (hh:mm).

Month	**Average overtime**	**Average undertime**
Jun	0:57	1:32
Jul	0:36	1:46
Aug	0:49	1:41
Sep	0:35	2:28
Oct	0:24	2:24
Nov	0:03	2:53
Dec	0:16	3:20
Grand Average	0:33	2:11

Table 4.3: Summary of total overtime and undertime incurred in 2021 (hh:mm).

Month	**Total overtime**	**Total undertime**
Jun	25:07	57:57
Jul	25:34	49:08
Aug	28:06	46:51
Sep	21:53	42:58
Oct	21:12	54:27
Nov	18:14	55:49
Dec	9:28	35:20
Grand Total	149:37	342:33

4.3 Problem Formulation

Given the challenge of excess undertime and overtime at the hospital, our primary objective is to minimize total overtime. We begin with the initial schedule created by surgeons and administrators, aiming to enhance it by rescheduling certain surgeries. The problem parameters and constraints are largely determined by the hospital setting, as we strive to minimize disruptions to existing preferences and workflows while adhering to the department's established policies and assumptions.

- Parameters
  - Operating Rooms (ORs)
    * 5 ORs.
    * Each OR ends either at 3:30pm or 5pm.
  - Surgeons
    * Surgeons do not share ORs.
    * Surgeons do not share patients.
  - Surgeries
    * Duration of each surgery is drawn from it's duration distribution.
    * Turnover time is drawn from a Gaussian distribution with a mean of 25 minutes and a standard deviation of 2.75 minutes.
  - Assumptions
    * Surgeries are known and can be scheduled one month in advance.
    * No cancelled surgeries.
    * No constraints on recover beds.
    * No emergency patients.
- Hard constraints
  - Non-overlapping surgeries
    * Each surgery must be assigned to exactly one OR.
  - Non-overlapping ORs

    * Surgeries scheduled in the same room must not overlap in time.
  - OR start time
    * Each OR starts its first surgery at 8am.
- Soft constraints
  - OR end time
    * Each OR ends its surgeries as close as possible to the scheduled end time of 3:30pm or 5pm.

4.4 Methodology

We aim to solve the surgical case scheduling problem by re-scheduling the existing plan of elective surgeries to a more preferable day and time in order to minimize overtime incurred for each day. Since the hospital department we work with does not accept emergency patients and has no known recovery bed availability constraints, these considerations are out of scope for the current study.

The solution approach to the scheduling problem involves several components. Firstly, we create surgical duration distributions based on the historical data to use for estimating the duration of each new surgical case. Secondly, we create a discrete-event simulation that initially helps us to better understand the problem at hand before it then serves as a solution approach for the problem. Finally, we create a heuristic optimization algorithm that utilizes the insights from the simulation and reschedules surgeries to days where each surgeon is least likely to run into overtime.

We begin with an overview of the surgical duration distribution experiments in Section 5. We experiment with different models to draw durations and select the one that produces the most accurate results. Next, Section 7 covers the inner workings of the discrete-event simulation (DES) model that takes an initial sequence of surgeries as input and produces an overview on each day's performance (including how much overtime and undertime each day and each room has incurred). Overall, the DES model is used for stochastic and deterministic evaluations, with the output either being used for optimization or evaluation purposes. Finally, Section 8 covers the proposed heuristic algorithm that optimizes the current schedule based on the metrics gained from the DES model. We rely on greedy heuristics with the objective to reduce the total overtime incurred by the schedule. The simulation optimization solution framework is presented in Fig. 4.7.

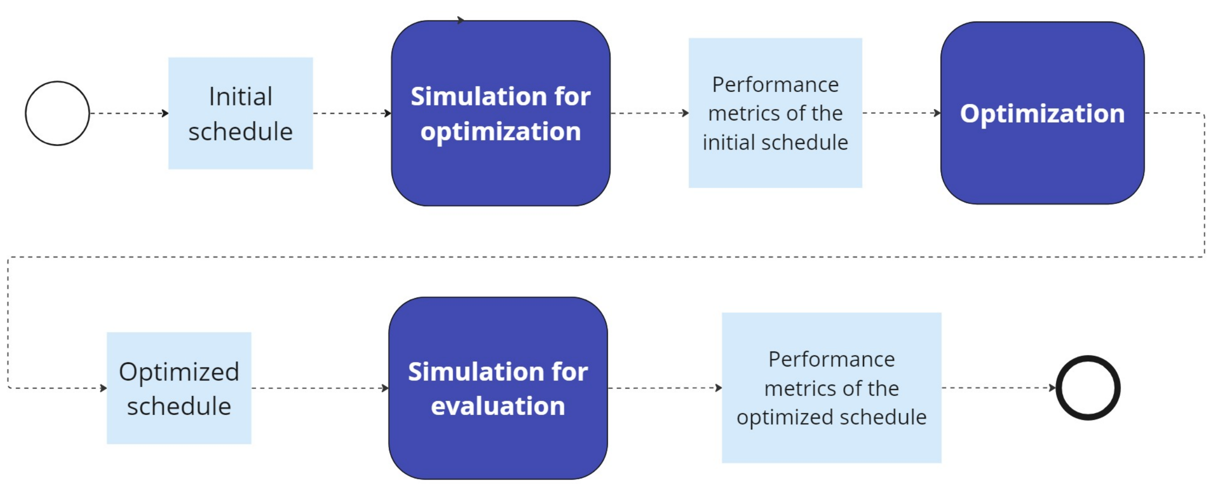


Figure 4.7: Simulation optimization architecture.

5. Surgical Duration Distributions

The simulation optimization approach presented in this thesis relies on drawing surgical durations from probability distribution functions (PDFs). These distributions represent the range of possible surgical durations based on historical data. By drawing surgical durations from the distributions, we simulate possible durations for a surgery based on the procedure. This method helps incorporate stochasticity into our simulation optimization model, capturing the inherent variability and uncertainty in surgery durations.

Surgical durations at large depend on a combination of patient, surgeon, and surgery factors. Several models have been experimented with to create distributions that can provide accurate surgical duration estimations for surgical cases. A number of key considerations were made in order to refine the final model.

The first consideration was the appropriate time frame of surgical data needed to be considered in order to strike a balance between historical depth and contemporary relevance. Hospitals are dynamic entities that change with time depending on new technological advances or certain regulations. As procedures change at the hospital, there might be more or less time needed to perform certain activities. For instance, a shift in the preferred anaesthesia technique within surgical teams can reduce or increase the time needed to put the patient to sleep or to wake them up after procedure ends, contributing to the overall surgical duration. The last 12 months of surgical data can be very reflective of the most up-to-date procedures and techniques used at the hospital, however, it might not provide enough samples for fitting the PDFs.

The second consideration was the choice of features that needed to be included in the data to enhance the accuracy of estimating surgical durations. There are two considerations here: the number of features to include and the granularity of these features. In terms of number of features, durations could be fitted not only based on their procedure types but also surgeon ID, making the distribution more accurately represent not only the uniqueness of the procedure but also the expertise and quickness of the surgeon performing the surgery. In terms of granularity, one particular example relates to the description of the procedure. Each procedure has a procedure category, class, and code which starts from a broad categorization of the procedure to a more specific identification of the procedure code. The main challenge that relates to both of these considerations is the curse of dimensionality. As the number of included features increases, the available data samples become sparse, leading to issues in accurately characterizing the underlying probability distribution [58]. This poses complications in fitting appropriate distribution functions and deriving meaningful insights, as the method not only suffers from estimation inaccuracies, but also becomes computationally intensive.

The third consideration was the choice of PDFs to use for fitting historical data. There exists a broad range of PDFs for continuous variables, with some having a better ability to represent certain distributions. Even if only the 10 most popular distributions are used, it can still be cumbersome to work with them. In the case of normal distributions - they only have two parameters, namely location and scale, which are widely referred to as mean and standard deviation. However, other distributions have additional parameters such as a shape which requires additional consideration when drawing distributions. As a result, accommodating distribution tails can be challenging as they are often highly skewed or long.

5.1 Duration distribution models

We experimented with three duration distribution models to test out various considerations, such as the appropriate time frame of the duration samples, the specific features to include, and tail trimming techniques. We used the python distfit package [59] to perform the fitting. Figure 5.1 summarizes the models we experimented with.

The first model includes all surgical durations grouped by procedure codes. For each procedure code, durations below the 5th percentile and above the 95th percentile are removed to ensure that rare or extreme values do not affect the mean or standard deviation of our distributions. Next, for each procedure code, we take the filtered duration samples and evaluate their fit against five different distributions: uniform, normal, generalized extreme value, Weibull, and Student's t. The best-fitting distribution is chosen based on a goodness

	Procedure code	Surgeon code	Tail trimming	Samples period
Model 1	✓	✗	Before fitting	Jan 2012 - May 2021
Model 2	✓	✗	After fitting	Jan 2012 - May 2021
Model 3	✓	✓	Before fitting	Last 30 durations

Figure 5.1: Duration distribution models

of fit test, specifically the lowest residual sum of squares (RSS). After all procedures are evaluated, we retrieve the duration distribution dataset of the model. Each row in the dataset represents a procedure code with the details of the best fitting distribution and its associated parameters. The parameters include the location, scale, shape, and size of the distribution.

To provide an example, let's take one procedure code and collect all duration samples from January 2012 and May 2021. Next, we fit the durations against five different distributions. Figure 5.2 shows a summary of how well the durations fit against the distributions. Based on the graph, the generalized extreme value (genextreme) distribution has the lowest RSS and is therefore chosen to represent the procedure's duration distribution. The PDF is shown in Figure 5.3, where we see the mean, the confidence interval, and the overall shape of the distribution with a heavy right skewness.

The second model is similar to the first model but involves a different tail trimming technique. In the first model, the tails are trimmed before the durations are fitted to the

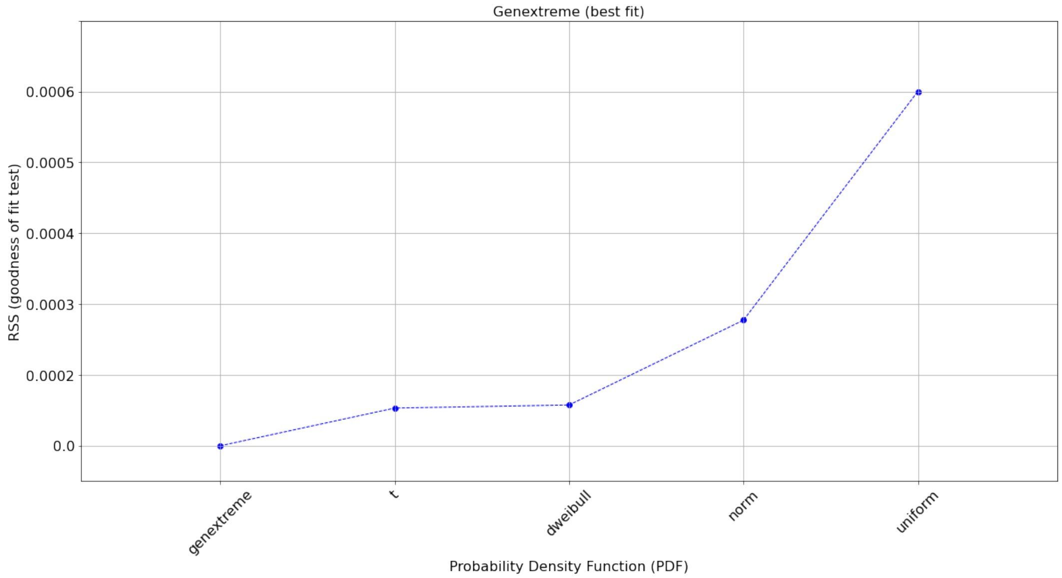


Figure 5.2: Summary of fitted theoretical distributions for one of the procedure codes.

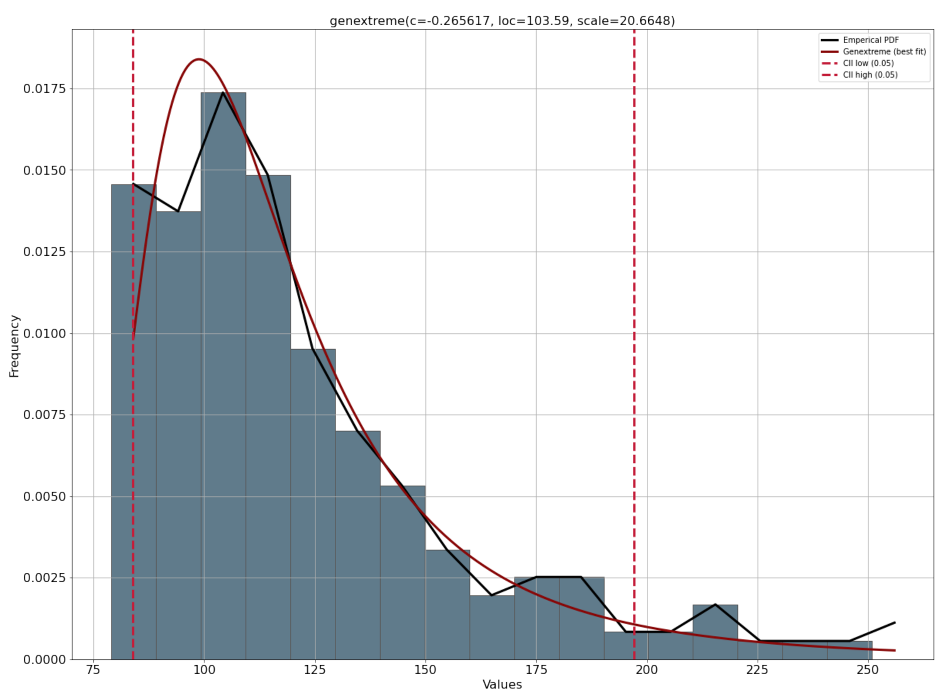


Figure 5.3: Distribution fit for one of the procedure codes.

distributions. However, in the second model, the durations are fitted to the distributions first. Then, based on the distribution parameters, we trim the tails. This approach allows us to customize the trimming technique depending on the distribution since each distribution typically has its own unique way of dealing with tails. After all durations under the procedure codes are trimmed, the revised duration samples are fitted onto 5 distributions and the duration distribution dataset of the second model is retrieved.

The third model is similar to the first model but all surgical durations are grouped by procedure codes, as well as by surgeon codes. In addition, the third model only collects and uses the last 30 samples of durations for each procedure and surgeon code combination instead of all samples between 2012 and 2022. This approach ensures that we only collect the recent samples of surgeries, increasing the chances to obtain distributions that are more accurate and relevant to the current practices. Similarly to model 1, the tails of the durations are trimmed before fitting and all durations below 5th percentile or above 95th percentile are removed. Then, durations are fitted onto 5 distributions. Finally, the duration distribution dataset of the third model is retrieved.

The process of fitting and evaluating our duration distributions is similar to how training and testing is done within the machine learning realm. First, we train our models (i.e.: fit surgical durations to the distributions). Second, we test each model's performance. This is why the duration distributions are fit on data between January 2012 and May 2021, whereas the data after May 2021 is used for our distribution performance evaluations as well as simulation and optimization work. Thus, we train on the historical data and test on the unseen data.

5.2 Performance evaluation

To evaluate the performance of the models, all three were run through the discrete-event simulation. The stochastic durations produced by the distributions were compared to the actual durations of each surgery that happened in real life. The testing was performed on June 2021 data. We took all surgeries during that month and generated duration samples for each surgery 1,000 times. These durations were determined based on the surgery's procedure code and, if applicable, the surgeon code. Mean absolute error (MAE) was used to measure the absolute difference between stochastic durations and actual durations. After analyzing the performance of surgical durations, we compared the average MAE value between the three models.

For some surgical cases, the procedure code that we needed wasn't present in the distribution table due to the procedure never happening in the past. To overcome this limitation, we employed strategy that would find the distribution of a similar procedure. The strategy involved searching for the distribution of the procedure code being performed by another surgeon or searching for a less granular description of the procedure. The latter works by leveraging the hierarchical nature of procedure descriptors: procedure codes are more granular descriptions of procedure classes which are more granular descriptions of procedure categories. For example, a procedure category of 1VA represents therapeutic interventions on the hip joint which is a very broad explanation. A procedure class of 1VA53 provides a more granular explanation and represents the implantation of internal device in the hip joint. A procedure code of 1VA53LAPNA provides the most granular explanation of the procedure and represents the implantation of internal device in the hip joint using an open approach with other specialized details. Let's say we can't find any distributions for a procedure code of 1VA53LAPNA. Then, we can start to iteratively truncate the procedure code by one character until we find a match in the distributions table. Suppose it takes us five iterations to find a suitable match for 1VC53. Now, if there are several distributions that match 1VC53, we take the distribution that is associated with the specific surgeon we're looking for. However, if there are still multiple distribution matches - we simply take the average of the returned surgical duration means, to avoid over-complicating the strategy. While this approach may result in a loss of granularity, it ensures we generate surgical durations based on the available knowledge.

After generating 1,000 surgical durations for each surgery in June 2021, we calculated the average MAE value for each model. The first and second distribution models achieved a similar MAE, with 29.73 and 30.40 minutes respectively. The third model demonstrated a slight improvement, resulting in a MAE of 22.70 minutes. We attribute the performance enhancement to the fact that the third model incorporates not only procedure codes but also surgeon codes, which makes generating surgical durations more accurate, preserving the skill level of each surgeon as well as their unique operating capabilities. As such, since the third model exhibits the best performance relative to the other models, it is selected for use in all subsequent simulation and optimization experiments.

6. Turnover Time Predictions

To maximize the accuracy of our simulation optimization approach, we attempted to predict turnover times. During brainstorming sessions, we identified several potential factors influencing turnover times, including surgery type, surgeon, time of day, and the number of surgeries concluding concurrently. The final prediction model yielded a mean absolute error of 10 to 11 minutes which was deemed to be insignificant by our clinical partners, so the prediction model wasn't used in the final discrete-event simulation model. However, the findings from this predictive modelling were utilized in the analysis section.

6.1 Linear regression

A linear regression model was chosen to predict turnover time, as it is well-suited for estimating the number of minutes it will take to prepare the room between surgeries. Linear regression fits a linear model by assigning weights to minimize the residual sum of squares. When multiple features are used to train the model, it is called a multiple linear regression (MLR) model. Table 6.1 presents the full list of features used.

As with duration distributions, time horizon is an important consideration when training the model. Two MLR models were trained: the first on data between 2012 to 2022 and the second on just the most recent data between 2020 to 2022.

Table 6.1: Features used to construct the multiple linear regression model

Feature type	**Feature names**
Numerical	Number of surgeries today, number of rooms in use today, hour of the day, number of concurrently ending surgeries
Categorical	Procedure category, anaesthesia technique, surgeon, concurrently ending surgery
Ordinal	Surgery number today, day of the week

Models were evaluated based on two metrics suitable for regression evaluations: R-squared (R^2) and mean absolute error (MAE). R^2, also known as the coefficient of determination, works by measuring the amount of variation of data predictions explained by the model. The higher the R^2 score, the more accurate the predictions are and the less variance there is in them. Generally, the value of R^2 lies in the range between 0 and 1 and, thus, is interpreted as a percentage value. MAE works by calculating the average absolute difference between the predicted values and the actual values. There are several reasons why the metric was chosen for our study. First, MAE is easy to interpret. The metric is expressed at the same scale as the target variable, making it easier for us to understand how well the MLR model predicts turnover time against actual values. Second, MAE treats all errors equally, making it robust to outliers (i.e.: scenarios where turnover times were longer than usual due to unforeseen circumstances). Third, MAE disregards direction of errors. Over-forecasting and under-forecasting of turnover times are both treated as equivalent errors.

6.2 Performance evaluation

For each of the models, the data was split into 80% training and 20% testing. In short, testing did not bring promising results. The first MLR model with 10 years worth of data (between 2012 and 2022) returned R^2 value of 39% and MAE of 10.86 minutes. The second MLR with 2 years worth of data (between 2020 and 2022) returned R^2 of 41% and MAE of 10.48. A prediction that is off by 10 - 11 minutes from the actual time is not an acceptable result for our clinical partners. In addition, the R^2 suggests that less than half of variation of predictions is explained by the model, which is not an acceptable result. As such, the predictions produced by the model could not be used in our stochastic simulation optimization model. Nevertheless, the feature importance extracted from the model proved to be of interest. It highlighted that certain procedures, anaesthesia techniques, and concurrently ending cases had a high correlation with turnover times between 2012-2022 and 2020-2022. Specifically, correlations related to concurrently ending cases proved to us that turnover time delays are caused from multiple surgeries ending at the same time, leading to a constraint on the environmental team, who are called to clean and prepare several rooms. The Problem Analysis Section 4.2.3 outlines these findings and clinical interpretations of the results.

7. Discrete-Event Simulation

A discrete-event simulation (DES) model was chosen for this project for several reasons. First, DES is event-driven, allowing us to model and evaluate the sequence and impact of events within the operating theatre. Second, DES is a powerful tool for assessing performance, providing a more robust analysis of schedule efficiency compared to simple historical analyses. Third, DES enables us to customize the logic of processes and metric collection, forming the basis for our optimization efforts presented in the next chapter. Consequently, the value of implementing a DES model in our project is twofold: it allows us to evaluate the performance of both original and optimized schedules, and it works in conjunction with the optimization algorithm to generate optimized solutions.

In hospital settings, simulation models can be quite extensive and consider upstream as well as downstream units. For instance, the hospital may be modelled as a dynamic system that includes pre-operative events (i.e.: surgeon consultation, patient registration), as well as post-operative events (i.e.: recovery bed assignment, patient discharge). However, in our case, we solely consider the inter-operative events. The events that trigger state transitions in our model are surgery start, surgery end, and activities of the cleaning crew. Figure 7.1 showcases the states that the room can be in depending on which event is happening. On a broader scale, Figure 7.2 illustrates the events and state transitions that happen in our model.

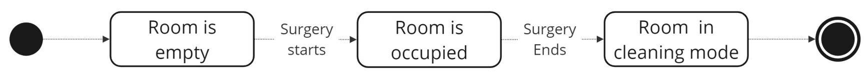


Figure 7.1: Room state transitions

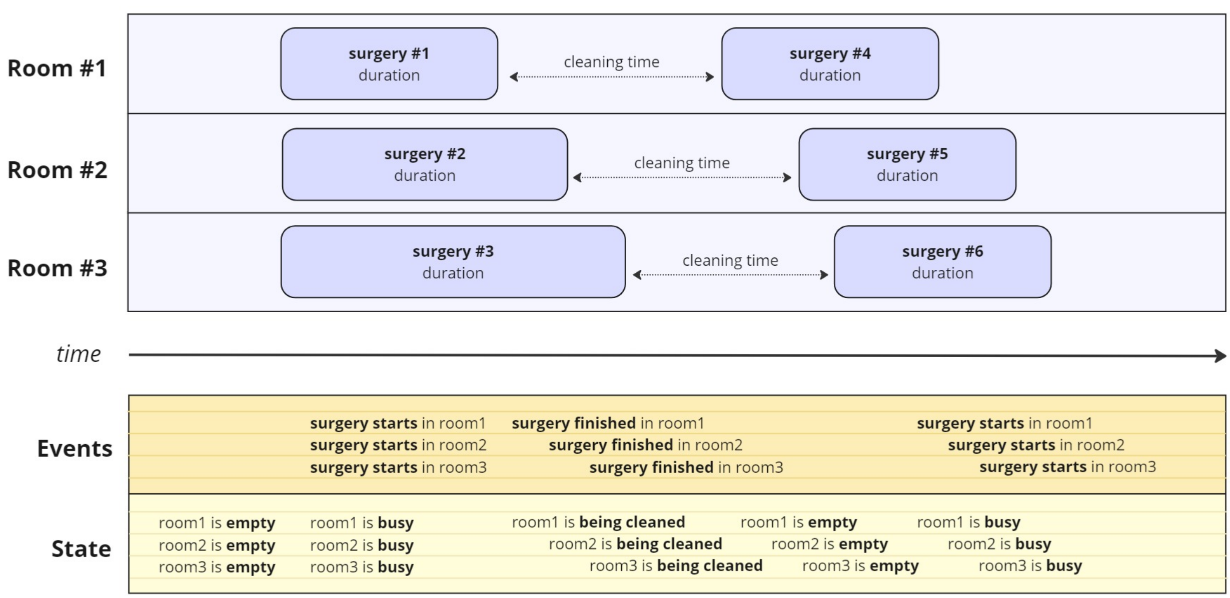


Figure 7.2: DES model flow

7.1 Simulation model setup

The input for our discrete-event simulation model is the sequence of surgeries that occurred in the hospital. We take each surgery, meaning each patient, one by one and dispatch them to a room once their assigned surgeon is available (has finished their previous surgery) and the room is available (has been prepared by the cleaning crew). Since the data we work with does not specify which room each surgeon operates in, we randomly assign each surgeon to a room at the beginning of the day. This follows the standard process at the hospital and ensures that no two surgeons share a room throughout the day. After the surgery finishes, the cleaning crew is dispatched into the room to prepare it for the next surgery. Since the data also doesn't have any information about expected turnover times or cleaning crew schedules, we estimate turnover time stochastically, drawing it from a probability distribution function. Each room begins its surgeries at 8am and is scheduled to end either at 3:30pm or 5pm, depending on the predetermined assignment.

Our DES operates in two modes: stochastic or deterministic, with outputs being used for optimization purposes and/or evaluation purposes respectively. The differentiation between the modes is presented in the next two sections, with the main algorithmic steps described in Algorithm 1. We built our discrete-event simulation model in Python 3.8.

7.2 DES for optimization

When the DES module is used for schedule optimization, surgery durations and turnover times are stochastic. Surgery durations get drawn from the probability distribution functions, depending on their surgical procedure code and surgeon performing the surgery. Turnover times are drawn from a Gaussian distribution with a mean of 25 minutes and standard

Algorithm 1 Simulation Algorithm

```
INITIALIZE
Load original schedule
Set mode to stochastic or actual
Declare simulated schedule dataframe
Declare simulated metrics dataframe
for each day in original schedule do
    for i in range 1,000 do
        Declare today's metrics
        Declare today's doctors, room assignment, room status, doctor status
        SIMULATE
        for each row in original schedule do
            Get surgery id, doctor id, procedure code
            Assign room to doctor id at surgery start time (current time)
            Set room status to occupied, doctor status to busy Calculate surgical duration:
            if mode == stochastic then
                Get duration from stochastic_duration(procedure code, surgeon id)
            else
                Get duration from original schedule
            end if
            Set surgery status end time (surgery start time + surgical duration)
            Set doctor status to available
            Calculate cleaning time, cleaning end time
            Set room status to available
            Append surgery details (i.e.: ID's, start and end times, duration) to simulated
            schedule dataframe
            COLLECT METRICS
            Calculate overtime and undertime
            Append overtime and undertime to the metrics
        end for
    end for
    Average out all metrics over 1,000 simulations
    Append averaged out metrics to simulated metrics dataframe
end for
Return simulated metrics dataframe
```

deviation of 2.75 minutes, the historical average turnover time at the hospital.

7.3 DES for evaluation

When the DES module is used for evaluation, surgery durations and turnover times can be either stochastic or deterministic, depending on the schedule being evaluated and purpose of the evaluation. There are two types of schedules: original (i.e.: the original sequence of surgeries) and reshuffled (i.e.: the optimized sequence of surgeries). In addition, there are two types of evaluations that we perform: evaluation of the current scheduling practices (as-is) and evaluation of the proposed scheduling optimization practises (to-be).

During the as-is evaluation, we compare the performances of the estimated schedule of surgeries to the actual schedule that happened. This gives us an idea of how much undertime and overtime was originally planned and how much undertime and overtime there actually was after the day has ended. The estimated schedule contains durations that were estimated by the hospital staff and relies on stochastic turnover times between surgeries in each room. Conversely, the actual schedule contains the actual timestamps of each surgery, thus it relies on surgical durations and turnover times that actually happened in real life. The parameters of the as-is evaluations are presented in Table 7.1.

Table 7.1: As-is schedule evaluations.

Schedule	**Type**	**Surgical Durations**	**Turnover Times**
Original	Estimated	Estimated	Stochastic
	Actual	Actual	Actual

During the to-be evaluation, we compare the performances of the original sequence of surgeries to the optimized sequence of surgeries. Here, the performances are evaluated in two ways: under a simulated environment and under an actual environment, as if the surgeries happened in real life.

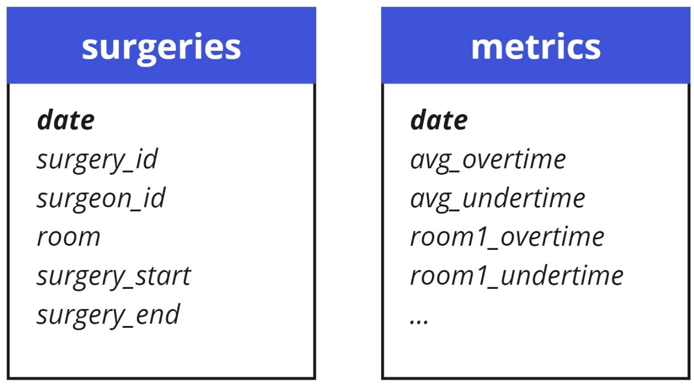
Under the simulated environment, the original and the reshuffled schedules rely on stochastic turnover times and stochastic durations. Whereas under the actual (assumed real life) environment, each schedule relies on stochastic turnover times but actual durations. The simulated environment allows us to compare the performances under a controlled environment, whereas the actual environment allows us to play out and compare the schedules as if they were to happen in real life. The parameters of the to-be evaluations are presented in Table 7.2.

Table 7.2: To-be schedule evaluations.

Schedule	**Type**	**Surgical Durations**	**Turnover Times**
Original	Actual	Actual	Stochastic
	Simulated	Stochastic	Stochastic
Reshuffled	Actual	Actual	Stochastic
	Simulated	Stochastic	Stochastic

7.4 Performance metrics

The performance metrics that get collected by the DES model are the same whether they are being collected for optimization or evaluation purposes. The difference only lies in the way those metrics are used. In optimization they're used for finding days that are likely to end up in overtime, while in evaluation they're used to compare performances of different schedules (i.e.: compare the amount of overtime incurred in the original schedule versus the optimized schedule).



(a) DES input: sequence of surgeries

(b) DES output: simulation metrics

Figure 7.3: Diagrams of tables used and produced by the simulation.

To calculate the simulation metrics, we first simulate each day in the schedule 1,000 times. Figure 7.3a describes the features that serve as input for the simulation. After 1,000 simulations, we are provided with an overview of the schedule performance in terms of total operating time, overtime, and undertime - all of which are averaged out across the simulation runs. Since each day is presented with uncertainty of surgical durations, simulating it numerous times provides us with an expected performance that is most likely to happen. After each day in the input schedule has been simulated, the averaged daily metrics are collected. Figure 7.3b describes the features (i.e.: metrics) that serve as output for the simulation. The full list and descriptions of the simulation metrics are presented below:

- Average overtime - Average minutes of overtime across all rooms in the day

- Average undertime - Average minutes of undertime across all rooms in the day
- R# Average overtime - Average minutes of overtime in Room #
  - If the room wasn't in use on any given date, the cell returns empty
- R# Average undertime - Average minutes of undertime in Room #
  - If the room wasn't in use on any given date, the cell returns empty

We differentiate between metrics for rooms and days in order to pinpoint not only the days with the most overtime/undertime but also specific rooms having the most effect on overtime/undertime.

8. Heuristic Optimization

During the brainstorming sessions with clinical partners, three candidate optimization techniques were proposed. One of them addresses the long-term Case Mix Problem (CMP) and two others address the short-term Surgical Case Scheduling Problem (SCSP).

The first technique (aimed at solving the CMP) focused on increasing the length of the operating day to better accommodate the delays and increase the number of surgeries happening in a day. By extending the length of the day, more surgeries could be accommodated within the open hours. However, the technique didn't seem appropriate for solving the presented problem of undertime versus overtime. The technique doesn't tackle the issue of OR utilization, rather it aims to increase throughput. Undeniably, once an appropriate optimization algorithm is identified, the opening hours of ORs could be re-considered depending on the surgical demand and the availability of resources to aid in dealing with surgical backlogs.

The second technique (aimed at solving the SCSP) focused on staggering the end times of surgeries, to make sure no two surgeries end concurrently. Figure 8.1 shows an example of a schedule with concurrently ending surgeries. When this happens, the cleaning crew is called to clean every room but they're only able to attend one OR at a time.

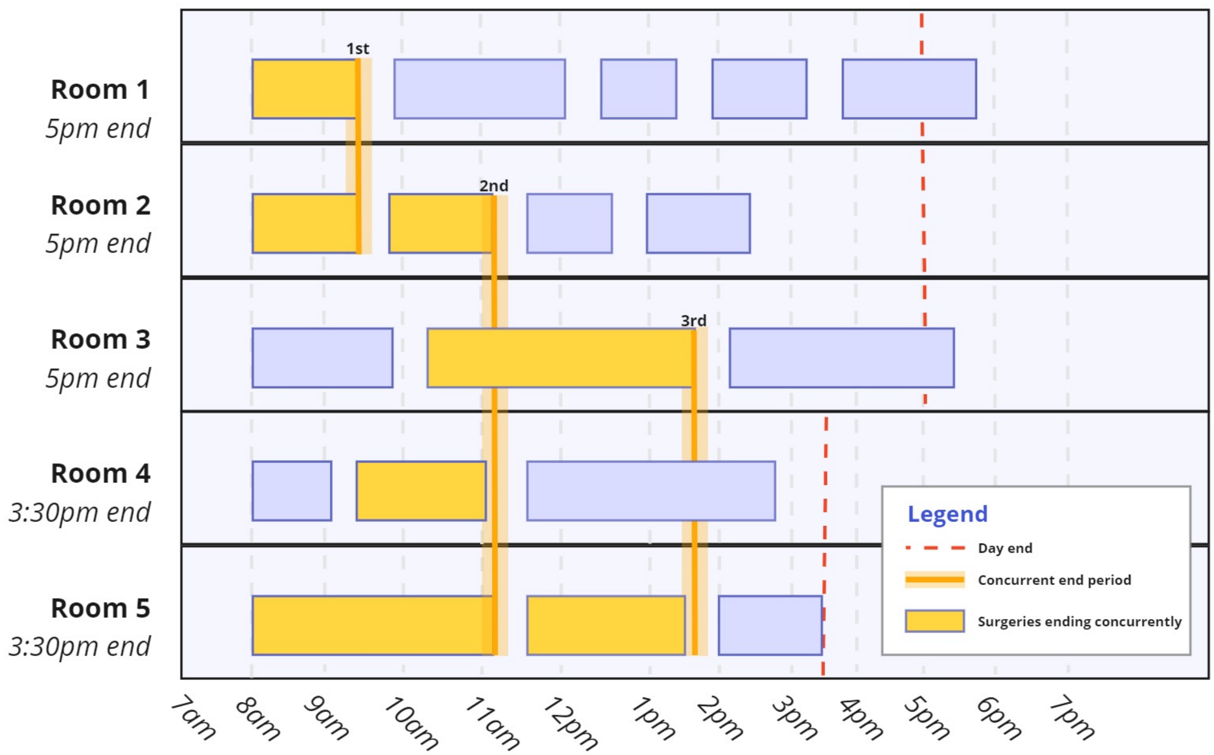


Figure 8.1: Schedule with multiple surgeries ending concurrently.

By staggering the end times of surgeries, the cleaning crew can have enough buffer to finish preparing one room and move on to the next room. Figure 8.2 shows an example of a potential revised schedule, where all surgeries end at least 15 minutes apart. This technique could effectively reduce the strain on cleaning crew and thus reduce the turnover time delays. However, from our analysis presented in Section 4.2, it was determined that while there's undeniably a strain on the cleaning crew when multiple surgeries end at the same time, the average delay of 2-3 minutes, caused by the cleaning crew delay, is not clinically significant to focus on and search for a solution at this time.

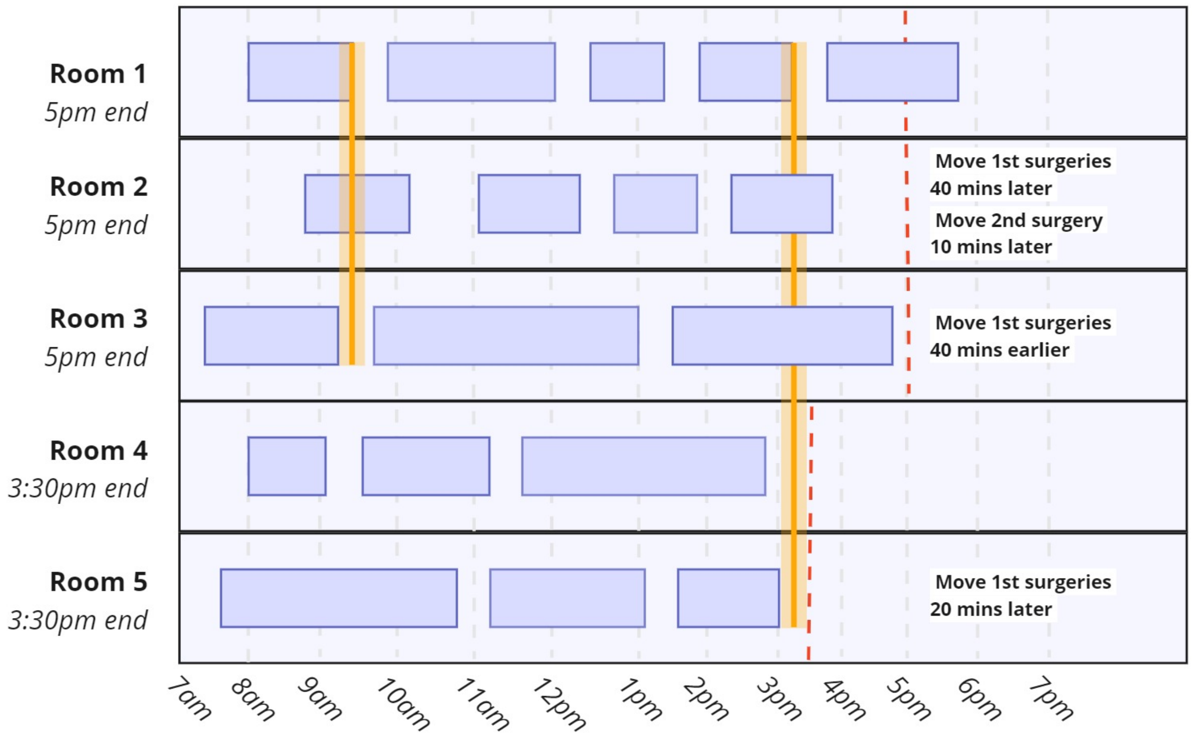


Figure 8.2: Optimized schedule where no two surgeries end concurrently.

The third technique (aimed at solving the SCSP) focused on switching surgeries between days within a monthly horizon. The objective is to minimize the amount of overtime each day accumulates by reshuffling surgeries from overtime days into undertime days. This technique was identified to have the best potential for optimizing the schedule comprehensively since it directly tackles the issue of overtime. Figure 4.3 in the analysis section showed that besides the presence of overtime, there's plenty of undertime in the current schedules. This means that with a more efficient scheduling technique, surgeries that end up in overtime could have been allocated into slots with undertime, and this could have reduced both the amounts of undesirable under or over utilization of the ORs. Thus, this thesis focuses on developing the reshuffling technique that would move certain surgeries from the overtime days into the undertime days, achieving a more optimal solution. The details of the reshuffling heuristic is presented in Section 8.1.

8.1 Reshuffling optimization models

We experimented with three reshuffling models: the "injection" model, which moves surgeries from overtime days to undertime days for the same surgeon; the "swapping" model, which exchanges long and highly variable surgeries from overtime days with short and less variable surgeries from undertime days; and the "combined" model, which blends both techniques based on their effectiveness in reducing total overtime.

Next, we compared the performances of the reshuffled schedules with those of the original schedule, evaluating them under both simulated and actual environments. Both environments relied on the same daily sequence of surgeries, but in the simulated environment, we used stochastic durations and turnover times, while in the actual environment, we relied on real durations and stochastic turnover times.

To conduct the simulation optimization of schedules, we used data from the last 7 months of 2021 surgical operations, comprising 1,415 surgeries. Our primary objective was to minimize overtime, so we selected data based on the availability of labels indicating the expected end times of operating rooms, either 3:30 pm or 5 pm. Additionally, we excluded the first 5 months of 2021 from our simulations due to COVID-19 restrictions, which influenced the number of surgeries performed during that period.

Before delving into the specifics of the optimization models, it's crucial to understand the overarching constraints imposed by the hospital. First, each surgeon's working days are fixed, meaning surgeries can only be rescheduled within days when the surgeon is scheduled to work, with no additional days added or removed. Second, the length of each surgeon's working day is predetermined as either a short or long day, maintaining this assumption while injecting or swapping surgeries. Third, the assignment of surgeons to patients is predetermined and cannot be modified. Our optimization objective is to minimize total monthly overtime while making limited changes to the original schedule.

Now, let's delve into the general structure of the optimization process, which remains consistent across all models. The input of the optimization models includes the original sequence of surgeries and associated metrics from the discrete-event simulation model. The model begins by iterating over each month present in the schedule and assumes that all of the surgeries in that month can be rescheduled to any other day within the month. To determine which surgeries in the selected month need to be reshuffled, we rank all days and their associated rooms based on the incurred overtime, then identify the surgeons responsible for those rooms. After selecting the top overtime room, we identify the surgeon that works in that room. Next, we select all other days that the surgeon works on and rank their days based on the incurred undertime. The day with the most undertime is selected as the best candidate. For each surgery in the selected overtime day, we draw 1,000 durations to determine average duration and standard deviation of each surgery. Subsequent steps vary depending on the selected model and the steps for each model is presented below. The overall pseudo-code for all models is presented in Algorithm 2.

Algorithm 2 Reshuffling algorithm

```
Load schedule dataset
Set mode to injection, swapping, or combined
for each month in schedule dataset do
    Initialize current overtime, current undertime, new overtime, new undertime
    while new overtime < current overtime do
        Select day, room, surgeon with the most overtime
        Find all other days that surgeon works on
        Select day and room with the most undertime
        for each surgery in overtime room do
            Draw 1,000 durations. Calculate mean and standard deviation
            if mode == injection then
                Go to Algorithm 3
            end if
            if mode == swapping then
                Go to Algorithm 4
            end if
            if mode == combined then
                Go to Algorithm 5
            end if
            Update new overtime and new undertime
        end for
    end while
end for
```

8.1.1 Injection model

The injection model aims to reduce overtime by moving the longest and most variable surgery from an overtime room to an undertime room. The model checks if the injection is feasible by comparing the duration of the surgery with the available undertime in the room. If the surgery fits into the available undertime, the injection is performed, and monthly metrics are updated. If the surgery doesn't fit into the available undertime, the model tries to inject the next longest and most variable surgery until a feasible injection is found or all surgeries in the overtime room are exhausted. The model then repeats this process for all overtime days and rooms before moving on to optimize the next month. The pseudo-code is presented in Algorithm 3.

Algorithm 3 Injection algorithm

```
Sort surgeries in the overtime room in descending order by mean and standard deviation
for each surgery in the overtime room do
    Select the longest and most variable surgery
    if injection is feasible then
        Inject long surgery to undertime room
        Break out of the loop
    end if
end for
```

8.1.2 Swapping model

The swapping model aims to reduce overtime by exchanging long and highly variable surgeries from overtime days with short and less variable surgeries from undertime days. After the overtime day and room is selected, the swapping model begins by drawing 1,000 durations for each surgery in the selected undertime day to determine average duration and standard deviation of each surgery. The model then selects the longest and most variable surgery from the overtime room and attempts to swap it with the shortest and least variable surgery from the undertime room. Feasibility of the swap is determined by analyzing its impact on

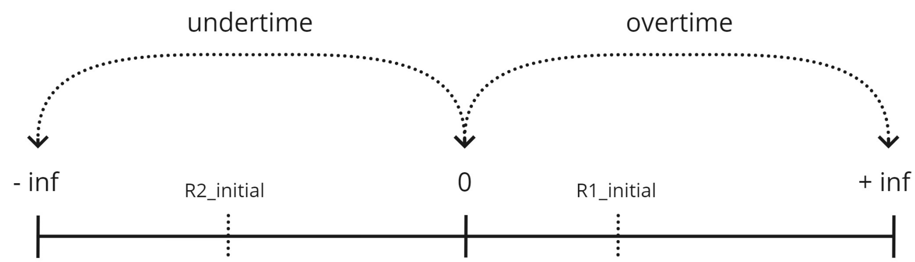


Figure 8.3: Overtime and undertime values on a continuum.

overtime and undertime in both rooms.

Let's say room 1 has overtime of $R1_{initial}$ and room 2 has undertime of $R2_{initial}$. If we imagine each room's timeline on a continuum, the value of overtime is going to be a positive one, whereas the value of undertime a negative one, with zero representing absence of over and under utilization of the OR, as illustrated in Figure 8.3. Once the swap is performed, $R1_{swap}$ is calculated as $R1_{initial}$ minus the duration of the removed long surgery S_L plus the duration of the added short surgery S_S. Similarly, $R2_{swap}$ is calculated by adding the duration of the long surgery S_L and subtracting the duration of the short surgery S_S. Refer to Equation 8.1.

- Calculate the feasibility of a swap:

$$
\begin{aligned}
\mathrm{R1_{swap}} &= \mathrm{R1_{initial}} - \mathrm{S_L} + \mathrm{S_S} \\
\mathrm{R2_{swap}} &= \mathrm{R2_{initial}} + \mathrm{S_L} - \mathrm{S_S}
\end{aligned}
\tag{8.1}
$$

If $R1_{swap}$ or $R2_{swap}$ results in an overtime greater than the original overtime value of

$R1_{initial}$, the swap is rejected, and the next longest surgery is attempted for swapping. The model continues to iterate through all surgeries in the overtime room until a feasible swap is found or until all surgeries in the overtime room are exhausted. The model then repeats this process for all overtime days and rooms before moving on to optimizing the next month. The pseudo-code is presented in Algorithm 4.

Algorithm 4 Swapping algorithm

```
for each surgery in undertime room do
    Draw 1,000 durations. Calculate mean and standard deviation
end for
Select surgery with shortest mean and standard deviation from undertime room
for surgery in overtime day do
    Select surgery with largest mean and standard deviation from overtime room
    if swap is feasible then
        Swap long overtime surgery with short undertime surgery
        Break out of the loop
    end if
end for
```

8.1.3 Combined model

The combined model blends the injection and swapping techniques and utilizes one or the other in each case based on their effectiveness in reducing total overtime. First and foremost, after the overtime day and room is selected, the combined model draws 1,000 durations for each surgery in the selected undertime day to determine average duration and standard deviation of each surgery. Then, the model evaluates the feasibility of both injection and swap techniques, considering the impact on cumulative overtime and undertime.

The feasibility of an injection is presented in Equation 8.2. It begins with a record of initial overtime and undertime in rooms $R1_{initial}$ and $R2_{initial}$. Then, the long duration surgery is added to the undertime room $R2_{initial}$ and is subtracted from the overtime room $R1_{initial}$. The resulting value of overtime/undertime in each room is denoted as $R1_{inject}$ and $R2_{inject}$. Similarly, the feasibility check of a swap returns the values of $R1_{swap}$ and $R2_{swap}$, as per the previously described Equation 8.1.

- Calculate the feasibility of an injection:

$$\begin{aligned} \mathrm{R1_{inject}} &= \mathrm{R1_{initial}} - \mathrm{S_L} \\ \mathrm{R2_{inject}} &= \mathrm{R2_{initial}} + \mathrm{S_L} \end{aligned} \tag{8.2}$$

Then, in order to decide which technique is better, we calculate the original cumulative overtime and undertime, as well as cumulative overtime and undertime after each model's reshuffle. These values are used to evaluate the total impact on overtime and undertime after performing one or the other technique. The cumulative overtime after swap CO_{swap} is calculated as a sum of $R1_{swap}$ and $R2_{swap}$, considering they're both positive. If only one of the $R1_{swap}$ or $R2_{swap}$ is positive, then only one is added to the sum. If none of the $R1_{swap}$ or $R2_{swap}$ is positive, then the sum is zero. The cumulative undertime after swap is calculated as a sum of $R1_{swap}$ and $R2_{swap}$, considering they're both negative. If only one of the $R1_{swap}$ or $R2_{swap}$ is negative, then only one is added to the sum. If none of the $R1_{swap}$ or $R2_{swap}$ is negative, then the sum is zero. A similar process is applied to calculate the cumulative overtime after injection CO_{inject} and cumulative undertime after injection CU_{inject}, shown in Equation (8.3) and (8.4), respectively.

- Calculating cumulative overtime after an injection:

$$CO_{\text{inject}} = \begin{cases} R1_{\text{inject}} + R2_{\text{inject}}, & \text{if } R1_{\text{inject}} \geq 0 \text{ and } R2_{\text{inject}} \geq 0, \\ R1_{\text{inject}}, & \text{if } R1_{\text{inject}} \geq 0 \text{ and } R2_{\text{inject}} < 0, \\ R2_{\text{inject}}, & \text{if } R1_{\text{inject}} < 0 \text{ and } R2_{\text{inject}} \geq 0, \\ 0, & \text{if } R1_{\text{inject}} < 0 \text{ and } R2_{\text{inject}} < 0. \end{cases} \tag{8.3}$$

- Calculating cumulative undertime after an injection:

$$CU_{\text{inject}} = \begin{cases} R1_{\text{inject}} + R2_{\text{inject}}, & \text{if } R1_{\text{inject}} < 0 \text{ and } R2_{\text{inject}} < 0, \\ R1_{\text{inject}}, & \text{if } R1_{\text{inject}} < 0 \text{ and } R2_{\text{inject}} \geq 0, \\ R2_{\text{inject}}, & \text{if } R1_{\text{inject}} \geq 0 \text{ and } R2_{\text{inject}} < 0, \\ 0, & \text{if } R1_{\text{inject}} \geq 0 \text{ and } R2_{\text{inject}} \geq 0. \end{cases} \tag{8.4}$$

If the cumulative overtime after an injection CO_{inject} is less than the cumulative overtime after a swap CO_{swap} and the cumulative overtime after an injection CO_{inject} is less than the original cumulative overtime $CO_{original}$, then the injection is accepted and vice versa. This ensures that whichever model is chosen reduces the original overtime better than the other model. If neither techniques sufficiently reduce overtime, the next longest surgery in the overtime room is checked again for both injection and swap feasibility. The model continues this process until a suitable technique is chosen or until all surgeries in the overtime room are exhausted. The model then repeats this process for all overtime days and rooms before moving on to optimizing the next month. The pseudo-code is presented in Algorithm 5.

Algorithm 5 Combined algorithm

```
for each surgery in undertime room do
    Draw 1,000 durations. Calculate mean and standard deviation
end for
Select undertime surgery with shortest mean and standard deviation
Check injection feasibility
Check swap feasibility
Select model that better improves cumulative overtime and cumulative undertime
```

To illustrate how the injection and the swapping models work, let's walk through the mechanism of the combined model. As an example, we take a scenario from June 2021. We identify the day and room with the most overtime (e.g., June 9) and the room with the most undertime (e.g., June 16). We assess the feasibility of both injection and swapping techniques and select the one that reduces the original overtime to a greater extent.

First, we check the feasibility of injecting a surgery. We take the longest surgery from the overtime room and attempt to fit it into the undertime room. If that doesn't work, we loop through other surgeries during the day and try to fit in any of them. In our case, we find it feasible to fit the shortest surgery from the overtime room into the undertime room as shown in Figure 8.4

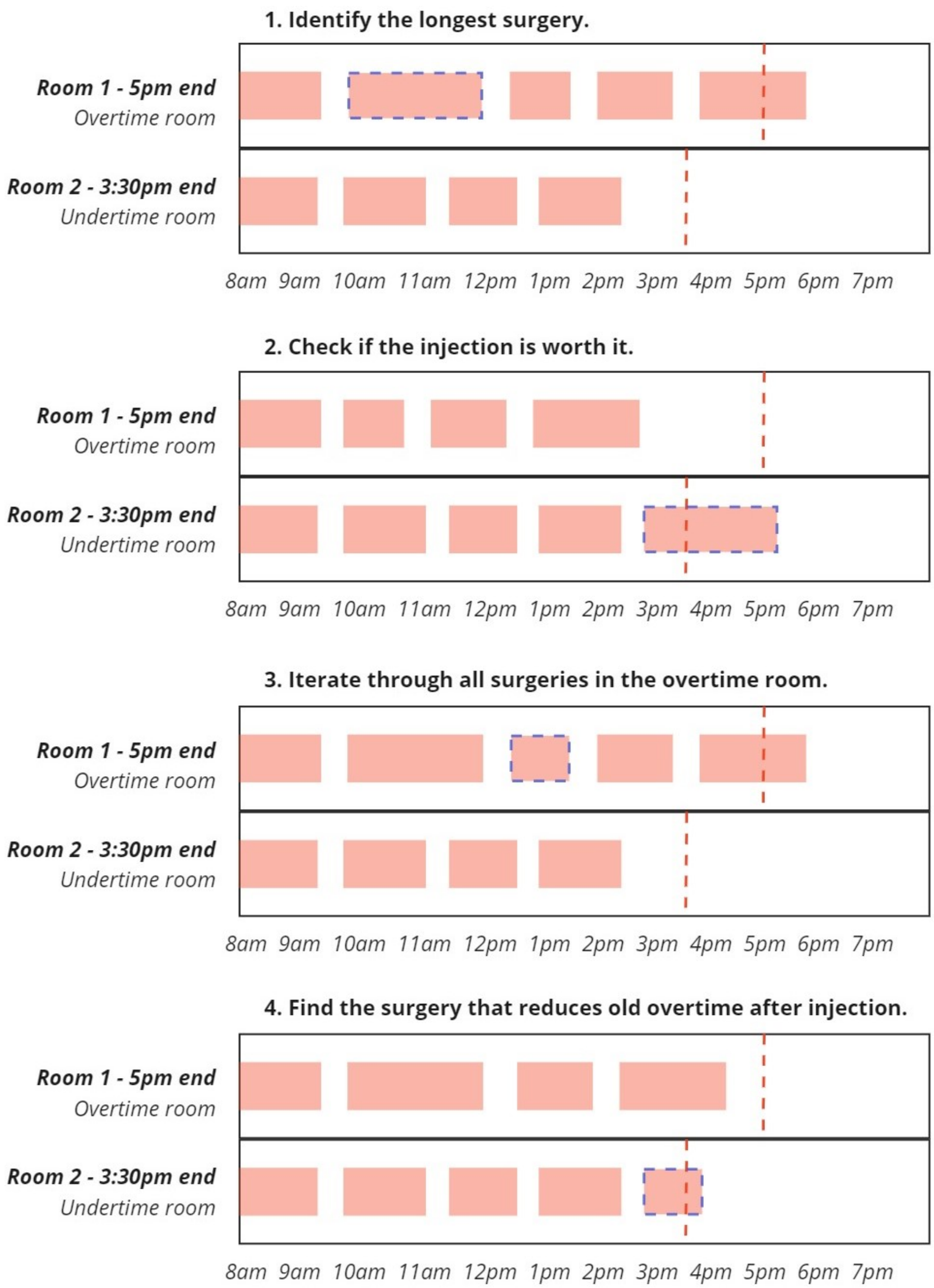


Figure 8.4: Injection model in action.

Second, we check the feasibility of swapping surgeries. We take the longest surgery from the overtime room and attempt to swap it with the shortest surgery from the undertime day, as shows in Figure 8.5.

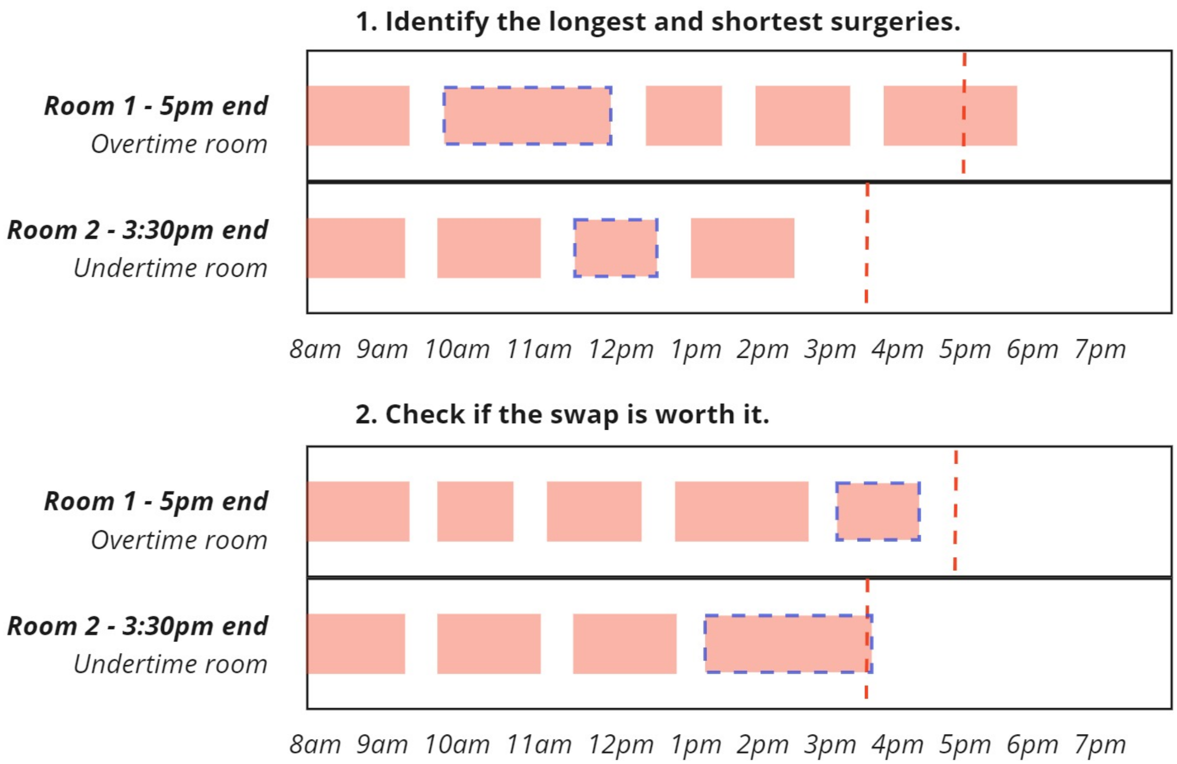


Figure 8.5: Swapping model in action.

Finally, as seen from the illustrations, both techniques are feasible (i.e. reduce the original cumulative overtime). Yet the swapping technique reduces overtime to a greater extent and would get chosen in this case.

8.2 Performance evaluation and results

To evaluate the performance of the models, each schedule that was generated by the optimization was run through the discrete-event simulation, collecting the performance metrics. The performances were first evaluated under the simulated environment (using stochastic durations), then under the actual environment (using actual durations).

8.2.1 Simulated environment comparison

Under the simulated environment, the swapping optimization model exhibited the poorest performance. It had the least ability to reduce the amount of overtime and undertime. Nonetheless, it was still able to achieve a 14.08% reduction of overtime compared to the original schedule and a 1.33% reduction of undertime. In contrast, the injection and combined models achieved better results. The injection model had a slightly better effect on overtime reduction, achieving a 43.95% reduction compared to 42.31% of the combined model. However, the combined model achieved a slightly better effect on undertime reduction, achieving a 4.48% reduction compared to 3.04% of the injection model. Table 8.1 compares performances of all models, both in terms of hours and percentage improvement.

Table 8.1: Total improvement of schedules under simulated durations (hh:mm).

Comparison	**Schedule**	**Total Overtime**	**Improvement**	**Total Undertime**	**Improvement**
Simulated durations	**Original**	71:52	-	278:37	-
	Injection	40:17	43.95%	270:08	3.04%
	Swapping	61:41	14.17%	274:54	1.33%
	Combined	41:30	42.26%	266:08	4.48%

In terms of the number of surgeries ending in overtime, we looked at the count of overtime ends for the day and percent of surgeries ending in overtime compared to total number of surgeries for the day. Figure 8.6 shows the mean percentage of surgeries ending in overtime. Both the injection model and the combined model were able to reduce the mean percentage of surgeries ending in overtime from 21% to 16%, whereas the swapping model had almost no effect.

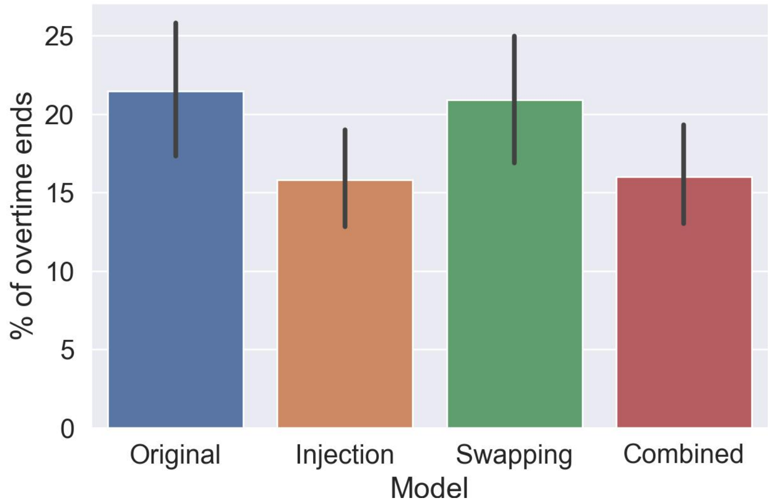


Figure 8.6: Percent of average overtime ends by model

Figure 8.7 shows the distribution of the average count of surgeries ending in overtime. The original model shows a relatively broad spread of values compared to the injection and combined models. This suggests that the injection and combined models not only reduce the mean number of overtime surgeries but also reduce the occurrence of more than 3 surgeries ending in overtime per day.

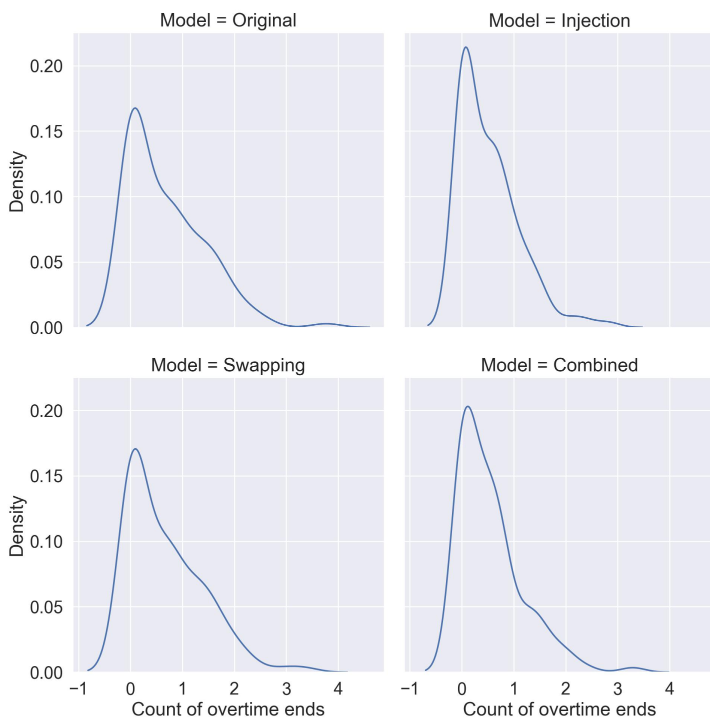
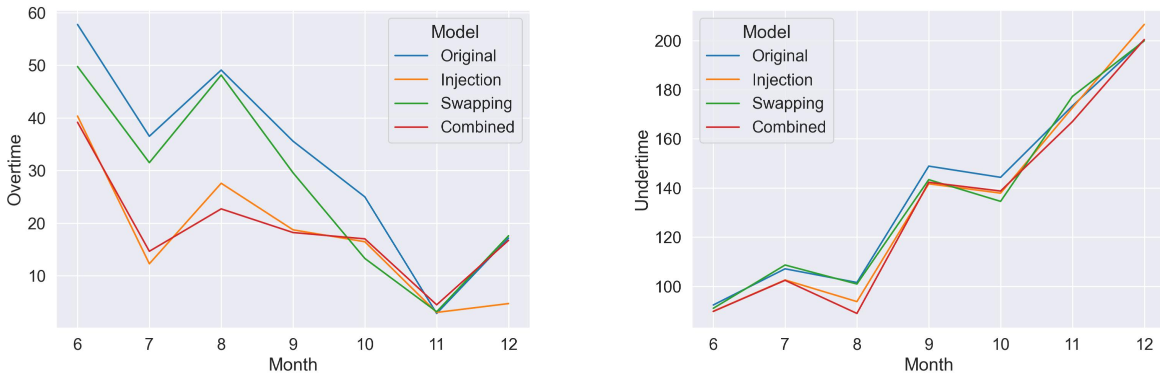


Figure 8.7: Distribution of average count of overtime ends by model

Table 8.2 shows the overtime and undertime achieved by each model grouped by month. Each model's performance is compared to the performance of the original schedule. Monthly overtime and undertime trends are shown in Figure 8.8. Certain months saw a higher reduction of overtime than others. November was the only month where all three models



(a) Average monthly overtime minutes. (b) Average monthly undertime minutes.

Figure 8.8: Comparison of simulated schedules by month.

exhibited a worse amount of overtime compared to original overtime value.

In addition to overtime and undertime performance evaluations, we also considered the number of reshuffles each model did per month, which is summarized in Table 8.3. The combined model performed less injections and less swaps compared to the injection and swapping models. This is attributed to the logic that in order to accept a swap or an injection, one of them must improve the original value of overtime better than the competing model. If one of the models produces overtime that's less that the competing model but that's greater than the original overtime value, then neither an injection nor a swap is performed.

Table 8.2: Comparison of simulated schedules by month using stochastic durations (hh:mm)

Month	Metric	Original	Injection	Swapping	Combined
June	Total overtime	20:12	14:07	17:24	13:42
	Overtime improvement	**-**	**30.12%**	**13.86%**	**32.18%**
	Total undertime	32:19	31:25	31:51	31:24
	Undertime improvement	**-**	**2.78%**	**1.44%**	**2.84%**
July	Total overtime	12:46	4:17	11:01	5:07
	Overtime improvement	**-**	**66.45%**	**13.71%**	**59.92%**
	Total undertime	37:28	35:53	38:01	35:50
	Undertime improvement	**-**	**4.23%**	**-1.47%**	**4.36%**
August	Total overtime	17:10	9:39	16:50	7:56
	Overtime improvement	**-**	**43.79%**	**1.94%**	**53.73%**
	Total undertime	35:32	32:48	35:19	31:07
	Undertime improvement	**-**	**7.69%**	**0.61%**	**12.43%**
September	Total overtime	10:04	5:18	8:22	5:09
	Overtime improvement	**-**	**47.34%**	**16.87%**	**48.80%**
	Total undertime	42:10	40:08	40:37	40:19
	Undertime improvement	**-**	**4.83%**	**3.69%**	**4.42%**
October	Total overtime	7:54	5:12	4:12	5:23
	Overtime improvement	**-**	**34.13%**	**46.82%**	**31.92%**
	Total undertime	45:42	43:39	42:36	43:56
	Undertime improvement	**-**	**4.46%**	**6.78%**	**3.86%**
November	Total overtime	0:51	0:54	0:57	1:20
	Overtime improvement	**-**	**-6.77%**	**-11.44%**	**-57.60%**
	Total undertime	52:00	51:47	53:10	50:05
	Undertime improvement	**-**	**0.44%**	**-2.23%**	**3.68%**
December	Total overtime	2:51	0:47	2:52	2:49
	Overtime improvement	**-**	**72.53%**	**-0.21%**	**1.37%**
	Total undertime	33:22	34:25	33:18	33:23
	Undertime improvement	**-**	**-3.13%**	**0.20%**	**-0.06%**

Overall, if we're comparing the models, the injection and combined models performed relatively similar in terms of reducing overtime, whereas the swapping model performs poorly compared to them. This can be attributed to the uniqueness of our data, where the original

Table 8.3: Number of injections and swap in each model.

Month	Metric	Injection	Swapping	Combined
June	Number of injections	4	-	4
	Number of swaps	-	6	0
July	Number of injections	9	-	8
	Number of swaps	-	11	3
August	Number of injections	5	-	6
	Number of swaps	-	11	2
September	Number of injections	5	-	3
	Number of swaps	-	9	3
October	Number of injections	2	-	0
	Number of swaps	-	4	2
November	Number of injections	1	-	2
	Number of swaps	-	3	0
December	Number of injections	3	-	1
	Number of swaps	-	0	0
Grand Total	**Number of injections**	**29**	**-**	**24**
	Number of swaps	**-**	**44**	**10**

schedule contains plenty of scheduled undertime, suggesting that injections of surgeries to undertime days are more likely to address the issue of overtime than swaps. However, injections alone might not be as effective at times. As such, the combination of injections with swaps enables us to not only address the underlying problem of undertime but also affect overtime to a greater extent, since swaps consider variability of surgery durations and help to redistribute surgeries based on length and historical variability of surgical durations.

The reason why there's still plenty of undertime even after reshuffling can be attributed to the fact that our optimization technique overlooks surgeons who don't work overtime. Since these surgeons are not considered for reshuffling by our models, the undertime amounts that those surgeons incur remains the same as in the original schedule. What's also interesting, not all surgeons experience undertime, or if they do, it's not enough undertime that would allow us to fit surgeries from overtime into. Our approach prioritizes minimal swaps and minimal schedule disruptions, therefore, we are not able to effectively addressed this issue with simple heuristics.

8.2.2 Actual environment comparison

Under actual environment comparison, we attempt to compare the actual schedule that happened to the optimized schedule that could've happened. Since neither our simulation, nor optimization components have seen the actual durations, in this comparison for both original and optimized schedules we use actual surgical durations that happened in real life and assume a stochastic turnover time drawn from a Gaussian distribution with a mean of 25 minutes. Table 8.4 compares performances of the models, both in terms of hours and percentage improvement.

Table 8.4: Total improvement of schedules under actual durations.

Comparison	**Schedule**	**Total Overtime hh:mm**	**Improvement**	**Total Undertime hh:mm**	**Improvement**
Actual durations	**Original**	57:46	-	240:16	-
	Injection	51:28	10.93%	239:13	0.44%
	Swapping	60:00	-3.84%	237:20	1.22%
	Combined	51:46	10.40%	234:15	2.50%

Once again, even under the inherently unpredictable conditions of the real-world surgical operations, the injection and combined models produce schedules that yield positive outcomes.

Under the actual environment, the injection model achieved an average of 10.93% reduction of overtime minutes and a 0.44% reduction of undertime minutes throughout the 7 months. The combined model achieved comparable results, with 10.40% reduction of overtime and 2.50% reduction of undertime. However, the swapping model did poorly. Compared to the original schedule, it incurred 7.33% more overtime and just 0.55% less undertime.

The reduction in overtime minutes by the injection and combined models highlights the practical applicability and effectiveness of these techniques in real-world healthcare settings. The potential reason for worse results with the swapping model is the fact that we've only tried swapping overtime with undertime days. The effects of reshuffling from overtime days to overtime days or from overtime days to days with very little undertime hasn't been studied in this work.

9. Conclusions and Future Work

Efficient operating room scheduling is important for keeping up with the growing demand for surgeries, as well as for staff and financial planning. This thesis demonstrates that a simulation optimization approach can be applied to solving the surgical scheduling problem. We show how the simulation component can be used for evaluating the impact of surgical duration variability on the overall schedules. In addition, we explain how the optimization component can be used for selecting surgeries that affect overtime and reshuffling them into days with undertime. While we found that taking surgeries from overtime days and injecting them into undertime days has the most effect on overtime, a combined approach that intelligently decides between injecting one surgery or swapping two surgeries shows a similar effect on overtime yet produces a slightly better effect on undertime. Since elective surgical scheduling is practically a dispatch system, the main issue remains to be the ability to predict surgical durations and handle their variances. Thus, future work needs to focus on improving the ability to estimate surgical durations that are more accurate compared to actual durations and experiment with other reshuffling techniques for achieving less overtime and higher utilization.

In general, this thesis work presents a contribution not only to the academic community but also to the practical healthcare management field. By bridging the gap between theoretical insights and real-world problems, we provide a data driven and tangible solution framework to the efficient surgical scheduling problem. The reliance on real-world scheduling data provided by the hospital enabled us to build the solution framework. It also allowed us to showcase the relevance of not only our theoretical results under the simulation environment, but also results under the presumed actual environment. In this environment, we attempted to mimic the real world by using actual surgical durations for each surgical case. Such an approach allowed us to contribute to one of the significant gaps that many of the theoretical papers exhibit - lack of real-world testing.

The development of the presented simulation optimization architecture is also a useful contribution for future research and innovation within the research group. Establishing the model has been necessary for developing a robust architecture that can be used to create and test optimization algorithms. This work will help to address the gap that exists currently in the field where there is limited unified ability to test optimization algorithms and compare results. The established architecture is also an example of the combined simulation optimization, which is a growing field due to advancements of computational powers and the need for innovative solutions.

There are several avenues for future research and refinement of the optimization algorithm.

Firstly, future work should enhance the optimization component by incorporating more scenarios and assumptions. For example, we should consider reshuffling surgeries for surgeons who either have no undertime and only work overtime or vice versa. This requires adjusting the monthly surgery throughput for each surgeon, either by reducing their workload to decrease overtime or increasing their workload to decrease undertime. Moreover, with such adjustments we could also change the scheduling horizon from a static monthly one to a variable rolling one, that could be on a weekly, bi-weekly, or monthly basis. This would enable us to more freely re-schedule surgeries considering a more flexible planning horizon.

Secondly, experimentation with multi-objective optimization is needed to study the effect of reshuffles not only on overtime but also undertime, OR utilization, throughput and waiting time. Incorporation of such objectives could help to develop more comprehensive and nuanced scheduling strategies that balance competing priorities and objectives.

Finally, ongoing collaboration between researchers, healthcare practitioners, and hospital administrators will be essential for driving innovation and implementing sustainable solutions in surgical scheduling. By fostering interdisciplinary partnerships and leveraging insights from both academia and industry, we can continue to advance the field of healthcare operations management and improve patient outcomes.

Bibliography

[1] Nov. 2023. URL: https://www.cihi.ca/en/national-health-expenditure-trends-2023-snapshot.

[2] Silviya Nikolova, Mark Harrison, and Matt Sutton. "The impact of waiting time on health gains from surgery: Evidence from a national patient-reported outcome dataset". In: *Health economics* 25.8 (2016), pp. 955–968.

[3] Nick D Clement et al. "The number of patients "worse than death" while waiting for a hip or knee arthroplasty has nearly doubled during the COVID-19 pandemic: A UK nationwide survey". In: *The Bone & Joint Journal* 103.4 (2021), pp. 672–680.

[4] The Lancet Rheumatology. "Too long to wait: the impact of COVID-19 on elective surgery". In: *The Lancet. Rheumatology* 3.2 (2021), e83.

[5] Health Canada. *A 10-year Plan To Strengthen Health Care*. Sept. 2004. URL: https://www.canada.ca/en/health-canada/services/health-care-system/health-care-system-delivery/federal-provincial-territorial-collaboration/

`first-ministers-meeting-year-plan-2004/10-year-plan-strengthen-health-care.html`.

[6] Justin Britt et al. "A stochastic hierarchical approach for the master surgical scheduling problem". In: *Computers & Industrial Engineering* 158 (2021), p. 107385.

[7] D Díaz-López et al. "A simulation-optimization approach for the surgery scheduling problem: a case study considering stochastic surgical times". In: *International Journal of Industrial Engineering Computations* 9.4 (2018), pp. 409–422.

[8] David Sier, Patrick Tobin, and Cathryn McGurk. "Scheduling surgical procedures". In: *Journal of the Operational Research Society* 48.9 (1997), pp. 884–891.

[9] Shing Chih Tsai, Yingchieh Yeh, and Chen Yun Kuo. "Efficient optimization algorithms for surgical scheduling under uncertainty". In: *European Journal of Operational Research* 293.2 (2021), pp. 579–593.

[10] Kyung Sung Jung et al. "Scheduling elective surgeries with emergency patients at shared operating rooms". In: *Production and Operations Management* 28.6 (2019), pp. 1407–1430.

[11] Hadhemi Saadouli et al. "A stochastic optimization and simulation approach for scheduling operating rooms and recovery beds in an orthopedic surgery department". In: *Computers & Industrial Engineering* 80 (2015), pp. 72–79.

[12] Rung-Chuan Lin, Mustafa Y Sir, and Kalyan S Pasupathy. "Multi-objective simulation optimization using data envelopment analysis and genetic algorithm: Specific application to determining optimal resource levels in surgical services". In: *Omega* 41.5 (2013), pp. 881–892.

[13] Atle Riise and Edmund K Burke. "Local search for the surgery admission planning problem". In: *Journal of Heuristics* 17 (2011), pp. 389–414.

[14] Wim Vancroonenburg, Pieter Smet, and Greet Vanden Berghe. "A two-phase heuristic approach to multi-day surgical case scheduling considering generalized resource constraints". In: *Operations Research for Health Care* 7 (2015), pp. 27–39.

[15] Shuwan Zhu et al. "Operating room planning and surgical case scheduling: a review of literature". In: *Journal of Combinatorial Optimization* 37 (2019), pp. 757–805.

[16] Sean Harris and David Claudio. "Current trends in operating room scheduling 2015 to 2020: a literature review". In: *Operations Research Forum*. Vol. 3. Springer. 2022, p. 21.

[17] Guoxuan Ma and Erik Demeulemeester. "A multilevel integrative approach to hospital case mix and capacity planning". In: *Computers & Operations Research* 40.9 (2013), pp. 2198–2207.

[18] Sebastian McRae and Jens O Brunner. "Assessing the impact of uncertainty and the level of aggregation in case mix planning". In: *Omega* 97 (2020), p. 102086.

[19] Sangdo Choi and Wilbert E Wilhelm. "On capacity allocation for operating rooms". In: *Computers & Operations Research* 44 (2014), pp. 174–184.

[20] Iman Rahimi and Amir H Gandomi. "A comprehensive review and analysis of operating room and surgery scheduling". In: *Archives of Computational Methods in Engineering* 28.3 (2021), pp. 1667–1688.

[21] Thomas Reiten Bovim et al. "Stochastic master surgery scheduling". In: *European Journal of Operational Research* 285.2 (2020), pp. 695–711.

[22] Jeroen M van Oostrum et al. "A master surgical scheduling approach for cyclic scheduling in operating room departments". In: *OR spectrum* 30 (2008), pp. 355–374.

[23] S Ayca Erdogan et al. "Surgery planning and scheduling". In: *Wiley encyclopedia of operations research and management science*. Citeseer, 2011.

[24] Alex Macario. "Are Your Hospital Operating Rooms "Efficient"?: A Scoring System with Eight Performance Indicators". In: *Anesthesiology* 105.2 (Aug. 2006), pp. 237–240.

[25] Tamas Fixler and James G Wright. "Identification and use of operating room efficiency indicators: the problem of definition". In: *Canadian Journal of Surgery* 56.4 (2013), p. 224.

[26] James M Magerlein and James B Martin. "Surgical demand scheduling: a review." In: *Health services research* 13.4 (1978), p. 418.

[27] Hesamoddin Tahami and Hengameh Fakhravar. "A literature review on combining heuristics and exact algorithms in combinatorial optimization". In: *European Journal of Information Technologies and Computer Science* 2.2 (2022), pp. 6–12.

[28] Davood Astaraky and Jonathan Patrick. "A simulation based approximate dynamic programming approach to multi-class, multi-resource surgical scheduling". In: *European Journal of Operational Research* 245.1 (2015), pp. 309–319.

[29] Jose M Molina-Pariente, Victor Fernandez-Viagas, and Jose M Framinan. "Integrated operating room planning and scheduling problem with assistant surgeon dependent surgery durations". In: *Computers & Industrial Engineering* 82 (2015), pp. 8–20.

[30] Belinda Spratt and Erhan Kozan. "A real-time reactive framework for the surgical case sequencing problem". In: *Flexible Services and Manufacturing Journal* 33.1 (2021), pp. 183–211.

[31] Mehdi Lamiri, Frédéric Grimaud, and Xiaolan Xie. "Optimization methods for a stochastic surgery planning problem". In: *International Journal of Production Economics* 120.2 (2009), pp. 400–410.

[32] Inês Marques, M Eugénia Captivo, and Margarida Vaz Pato. "Scheduling elective surgeries in a Portuguese hospital using a genetic heuristic". In: *Operations Research for Health Care* 3.2 (2014), pp. 59–72.

[33] Wei Xiang, Jiao Yin, and Gino Lim. "An ant colony optimization approach for solving an operating room surgery scheduling problem". In: *Computers & Industrial Engineering* 85 (2015), pp. 335–345.

[34] Azadeh Farsi, S Ali Torabi, and Mahdi Mokhtarzadeh. "Integrated surgery scheduling by constraint programming and meta-heuristics". In: *International Journal of Management Science and Engineering Management* 18.4 (2023), pp. 292–304.

[35] Shan Wang et al. "Surgery scheduling in the presence of operating room eligibility and dedicated surgeon: an adaptive composite dispatching method". In: *International Journal of Production Research* 61.6 (2023), pp. 1866–1881.

[36] Dinh-Nguyen Pham and Andreas Klinkert. "Surgical case scheduling as a generalized job shop scheduling problem". In: *European Journal of Operational Research* 185.3 (2008), pp. 1011–1025.

[37] Gonçalo Figueira and Bernardo Almada-Lobo. "Hybrid simulation–optimization methods: A taxonomy and discussion". In: *Simulation Modelling Practice and Theory* 46 (2014), pp. 118–134.

[38] Yao Xiao and Reena Yoogalingam. "A simulation optimization approach for planning and scheduling in operating rooms for elective and urgent surgeries". In: *Operations Research for Health Care* 35 (2022), p. 100366.

[39] Feng Liang, Yuanyuan Guo, and Richard YK Fung. "Simulation-based optimization for surgery scheduling in operation theatre management using response surface method". In: *Journal of medical systems* 39 (2015), pp. 1–11.

[40] Paolo Landa et al. "A hybrid optimization algorithm for surgeries scheduling". In: *Operations Research for Health Care* 8 (2016), pp. 103–114.

[41] Mehdi Lamiri et al. "A stochastic model for operating room planning with elective and emergency demand for surgery". In: *European Journal of Operational Research* 185.3 (2008), pp. 1026–1037.

[42] Dirk P Kroese et al. "Why the Monte Carlo method is so important today". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 6.6 (2014), pp. 386–392.

[43] Joseph M O'Brien Antognini, Joseph F Antognini, and Vijay Khatri. "How many operating rooms are needed to manage non-elective surgical cases? A Monte Carlo simulation study". In: *BMC health services research* 15 (2015), pp. 1–9.

[44] Raul Pulido et al. "Optimization methods for the operating room management under uncertainty: Stochastic programming vs. decomposition approach". In: *Journal of Applied Operational Research* 6.3 (2014), pp. 145–157.

[45] Aazad Abbas et al. "Machine learning using preoperative patient factors can predict duration of surgery and length of stay for total knee arthroplasty". In: *International journal of medical informatics* 158 (2022), p. 104670.

[46] Beiqun Zhao et al. "A machine learning approach to predicting case duration for robot-assisted surgery". In: *Journal of medical systems* 43 (2019), pp. 1–8.

[47] Jianzhe Luo, Vidyadhar G Kulkarni, and Serhan Ziya. "Appointment scheduling under patient no-shows and service interruptions". In: *Manufacturing & Service Operations Management* 14.4 (2012), pp. 670–684.

[48] Brian Denton and Diwakar Gupta. "A sequential bounding approach for optimal appointment scheduling". In: *IIE transactions* 35.11 (2003), pp. 1003–1016.

[49] Taghi Khaniyev, Enis Kayış, and Refik Güllü. "Next-day operating room scheduling with uncertain surgery durations: Exact analysis and heuristics". In: *European Journal of Operational Research* 286.1 (2020), pp. 49–62.

[50] Elliott N Weiss. "Models for determining estimated start times and case orderings in hospital operating rooms". In: *IIE transactions* 22.2 (1990), pp. 143–150.

[51] J Theresia Van Essen et al. "Decision support system for the operating room rescheduling problem". In: *Health care management science* 15 (2012), pp. 355–372.

[52] Babak Akbarzadeh et al. "The re-planning and scheduling of surgical cases in the operating room department after block release time with resource rescheduling". In: *European Journal of Operational Research* 278.2 (2019), pp. 596–614.

[53] Masoud Eshghali et al. "Machine learning based integrated scheduling and rescheduling for elective and emergency patients in the operating theatre". In: *Annals of Operations Research* 332.1 (2024), pp. 989–1012.

[54] Robert W. Allen, Kevin M. Taaffe, and Gilbert Ritchie. "Surgery rescheduling using discrete event simulation: A case study". In: *Proceedings of the Winter Simulation Conference 2014*. 2014, pp. 1365–1376. DOI: `10.1109/WSC.2014.7019991`.

[55] Francisco Ballestín, Ángeles Pérez, and Sacramento Quintanilla. "Scheduling and rescheduling elective patients in operating rooms to minimise the percentage of tardy patients". In: *Journal of Scheduling* 22 (2019), pp. 107–118.

[56] Wil Van Der Aalst. "Process mining: Overview and opportunities". In: *ACM Transactions on Management Information Systems (TMIS)* 3.2 (2012), pp. 1–17.

[57] mindzie, inc. *mindzie, inc.* June 10, 2023. URL: `https://mindzie.com`.

[58] Naomi Altman and Martin Krzywinski. "The curse (s) of dimensionality". In: *Nat Methods* 15.6 (2018), pp. 399–400.

[59] Erdogan Taskesen. *distfit is a python library for probability density fitting*. Version 1.4.0. 2020. URL: `https://erdogant.github.io/distfit`.