Università degli Studi di Pavia
Dipartimento di Matematica
Corso di Laurea Magistrale in Matematica

UNIVERSITÀ
DI PAVIA

Apprendimento guidato dall'ottimizzazione per politiche di pianificazione in tempo reale delle sale operatorie

Optimization-Guided Learning for Online Operating Room Scheduling Policies

Tesi di Laurea Magistrale in Matematica

Relatore (Supervisor):
Davide Duma

Tesi di Laurea di:
Chiara Marinelli
Matricola 561083

Anno Accademico 2025-2026

A Mamma e Papà,
per aver gonfiato le mie vele.
Se sono arrivata fino a qui senza
mettere mano ai remi, è grazie a voi.

A Laura,
per aver navigato accanto a me.
Sarà ancora meglio se continueremo ad essere
su due barche diverse: gli artisti sono difficili da capire.

Abstract

Operating room schedules are defined before the actual conditions under which they will be carried out are known. Variability in surgical durations and stochastic non-elective patient arrivals reduce the capacity available for scheduled activity. During the operating day, it may consequently become necessary to decide whether to perform the next elective surgery, accepting the risk of additional overtime, or to cancel it.

This thesis develops an optimization-guided machine learning framework that transfers offline optimization decisions to a fast online policy. Weekly elective schedules are constructed through a two-stage stochastic integer programming model under patient-centered and facility-centered objectives. A perfect-information MILP oracle then generates execute/cancel decisions under Monte Carlo realizations of surgical durations and non-elective demand. These decisions are used as labels for classifiers trained only on information observable at the corresponding decision time.

The selected policy is an XGBoost classifier with execution threshold $\tau = 0.12$, evaluated in a closed-loop discrete-event simulation. Compared with an *always-execute* policy, it reduces mean extra overtime by approximately 55% under facility-centered schedules and 49% under patient-centered schedules. Compared with a conservative *expected-capacity* rule, it preserves more elective activity under facility-centered schedules and more patient priority under patient-centered schedules, while producing fewer cancellations, although at the cost of higher overtime. Under facility-centered schedules, this corresponds to about 225 additional minutes of executed elective activity per week.

The simulation also shows that approximately one quarter of the canceled elective surgeries do not result from a separate decision by the classifier. Indeed, canceling one surgery also leads to the cancellation of the subsequent procedures in the same operating room sequence. Consequently, predictive metrics computed at the level of individual decisions alone do not fully capture the operational impact of the policy.

The computational results show that integrating stochastic programming, supervised learning, and simulation can support real-time operational decision-making while explicitly accounting for the trade-off between elective activity performed, patient priorities, and overtime utilization.

Riassunto

I piani delle sale operatorie vengono definiti prima che siano note le condizioni effettive in cui dovranno essere eseguiti. La variabilità delle durate chirurgiche e gli arrivi stocastici dei pazienti non elettivi possono ridurre la capacità disponibile per l'attività programmata. Durante la giornata operatoria può quindi rendersi necessario decidere se eseguire il successivo intervento elettivo, accettando il rischio di ulteriore straordinario, oppure cancellarlo.

Questa tesi sviluppa un approccio di machine learning guidato dall'ottimizzazione che trasferisce decisioni ottenute mediante ottimizzazione offline a una rapida policy decisionale online. I piani settimanali degli interventi elettivi vengono costruiti mediante un modello di programmazione stocastica intera a due stadi, secondo obiettivi patient-centered e facility-centered. Un modello MILP oracle a informazione perfetta genera successivamente decisioni di esecuzione/cancellazione sotto realizzazioni Monte Carlo delle durate chirurgiche e della domanda non elettiva. Tali decisioni vengono utilizzate come etichette per classificatori addestrati esclusivamente su informazioni osservabili al momento della corrispondente decisione.

La policy selezionata è un classificatore XGBoost con soglia di esecuzione $\tau = 0.12$, valutato mediante una simulazione a eventi discreti a ciclo chiuso. Rispetto a una policy *always-execute*, essa riduce lo straordinario aggiuntivo medio di circa il 55% nei piani facility-centered e del 49% nei piani patient-centered. Rispetto a una regola conservativa *expected-capacity*, preserva una maggiore attività elettiva nei piani facility-centered e una maggiore priorità complessiva dei pazienti nei piani patient-centered, producendo al tempo stesso un numero inferiore di cancellazioni, sebbene a fronte di un maggiore ricorso allo straordinario. Nei piani facility-centered, ciò corrisponde a circa 225 minuti aggiuntivi di attività elettiva eseguita per settimana.

La simulazione mostra inoltre che circa un quarto delle cancellazioni degli interventi elettivi non deriva da una decisione distinta del classificatore. Infatti, la cancellazione di un intervento comporta anche la cancellazione degli interventi successivi nella stessa sequenza di sala operatoria. Di conseguenza, le sole metriche predittive calcolate a livello delle singole decisioni non descrivono completamente l'impatto operativo della policy.

I risultati computazionali mostrano che l'integrazione tra programmazione stocastica, apprendimento supervisionato e simulazione può supportare decisioni operative in tempo reale, mantenendo esplicito il compromesso tra attività elettiva eseguita, priorità dei pazienti e utilizzo dello straordinario.

Contents

Chapter 1

Introduction

Operating rooms are among the most resource-demanding and strategically vital units in a modern hospital. Their activity requires the synchronized availability of specialized surgical teams, anesthesiologists, nursing personnel, advanced medical equipment, and downstream resources such as recovery and intensive-care capacity. Operating room expenses constitute a large part of total hospital budgets [BD07; CDB10]. Ensuring a high utilization of surgical capacity is therefore essential not only from an organizational and financial standpoint, but also to maintain equitable patient access to care and mitigate the social burden of growing waiting lists.

Despite the necessity of careful long-term planning, surgical schedules are inevitably constructed under uncertainty, whereas their execution unfolds in a stochastic and highly changing environment. Actual surgical durations frequently deviate from their expected values, delays propagate across consecutive procedures within the same operating room, and non-elective patients demand immediate access to common resources [ABK25].

This operational reality highlights a trade-off between decision quality and computational speed. Strict mathematical optimization models can reflect complex operational constraints and competing objectives in great detail, making them exceptionally well-suited for delivering high-quality decisions offline [BD07; CDB10]. Repeatedly solving a large mixed-integer programming problem whenever the operating-room state changes may, however, be impractical when decisions must be produced in real time. At the opposite methodological extreme, simple heuristic rules require little computing time, but they produce sub-optimal solutions.

This thesis investigates an intermediate approach: mathematical optimization is used offline to generate high-quality execute/cancel decisions, while supervised machine learning learns to approximate these decisions from information available at the corresponding online decision point.

The offline framework begins with weekly elective schedules constructed through a two-stage stochastic integer programming model under patient-centered and facility-centered objectives. Surgical-duration uncertainty is represented by Monte Carlo scenarios, that is, adopting a Sample Average Approximation (SAA) approach [Ber+25]. Scenario reduction is performed over the complete case mix, and K-means clustering

is used to obtain a smaller weighted set of representative realizations. Stochastic non-elective demand is subsequently introduced as competing workload for the same operating room capacity, following the shared-capacity perspective studied by Duma and Aringhieri [DA19].

After the offline schedules have been developed, an ex-post Mixed-Integer Linear Programming (MILP) oracle is solved for each simulated scenario. This oracle makes use of the full realized scenario and identifies the execute/cancel decisions that would be favored by the perfect-information optimization model. The supervised training labels are derived from the oracle's responses. The feature vectors, moreover, supplied to the classification models include only those details that were observable up to the relevant decision point: future non-elective arrivals and the unobserved actual durations are completely omitted, along with the oracle-specific variables. The learning problem is thus not set up as involving the prediction of an uncertain parameter to be fed into a later optimizer, but rather as directly applying offline mathematical foresight to an online classification policy.

The methodological framework described above is independent of the specific data source used for its computational evaluation; the benchmark data are used to parameterize and test the proposed approach rather than to define it. The empirical and numerical evaluation of this thesis relies on specialty-specific duration distributions derived from the *German Operating Room Benchmarking Initiative*, a large standardized repository of peri-operative process data described by Korzhenevich and Zander [KZ24]. Synthetic case-mix instances are generated across three different surgical specialties–General Surgery, Otolaryngology, and Gynecology & Obstetrics–providing a heterogeneous testing ground for the optimization, learning, and simulation pipelines. We evaluate twenty-nine candidate configurations among Decision Trees, Logistic Regression, Random Forests, and XGBoost. Rather than depending solely on classification accuracy, model selection is governed by a joint criterion that weighs predictive performance against the actual operational outcomes of the resulting policies [Man+24].

The final selected policy–an XGBoost classifier operating at the decision threshold $\tau = 0.12$ selected through the joint predictive and operational validation procedure–is embedded in an event-driven Discrete-Event Simulation (DES) environment [JJS99; FB22]. In the closed-loop setup, surgical durations are stochastic, and non-elective arrivals unfold chronologically. The resulting policy occupies an intermediate region of the elective-service–overtime trade-off, connecting offline optimization with a decision rule that can be evaluated directly during schedule execution.

The remainder of the thesis is organized as follows. Chapter 2 provides a survey of the literature, highlighting the research gap and the positioning of this study. Chapter 3 defines the online operating-room decision problem and presents the two offline optimization models supporting the proposed framework. Chapter 4 describes the solution method for the weekly scheduling problem, including Sample Average Approximation and scenario reduction. Chapter 5 presents the chronological reconstruction of online-safe decision states and the optimization-guided learning procedure. Chapter 6 introduces the discrete-event simulation used for closed-loop policy evaluation.

Chapter 7 describes the empirical data, synthetic instance generation, and experimental setup. Chapter 8 reports the computational results, while Chapter 9 discusses the main findings, limitations, and directions for future research.

Chapter 2

Literature Review

Operating room decision making spans two closely connected problems. Before execution, capacity must be allocated, and patients must be scheduled under uncertain durations and future demand. During execution, the same uncertainty becomes progressively observable and may require the original schedule to be adapted. The literature has addressed these problems through stochastic optimization, adaptive scheduling, machine learning, and simulation, but these methodological families differ in where computation is performed and in how uncertainty enters the decision process.

The review below is organized around this distinction. It first examines offline scheduling under uncertainty and real-time adaptation, then considers optimization-based and learning-based approaches, and finally turns to simulation as a way of evaluating sequential policies under endogenous state evolution.

2.1 Operating Room Planning and Scheduling under Uncertainty

Operating room planning and scheduling is a well-established area of healthcare operations research because surgical services consume substantial hospital resources and require the simultaneous coordination of operating rooms, surgeons, anesthesiologists, nursing staff, equipment, and downstream capacity [CDB10]. The latest reviews show that the field is still very active and heterogeneous, with optimization models dealing with a wide variety of objectives, constraints, planning horizons, and sources of uncertainty [ABK25].

The literature typically draws a distinction between various levels of decision-making [Hul+12]. Decisions at the strategic level relate to long-term capacity and infrastructure, for example, by determining the number of operating rooms and the amount of staffing. Tactical decisions involve the allocation of this capacity among different specialties and recurring blocks, generally through the use of a Master Surgical Schedule. Operational decisions then decide which patients are to be treated, how they are assigned to operating rooms, and in what order the procedures are carried out. Even though these levels are in

concept separate, the decisions taken at one level have an impact on the ones available at the following stages. Melo and Fogliatto's most recent systematic review highlights the increasing interest in models that explicitly integrate multiple decision levels, while at the same time pointing out the extra computational complexity that results from such integration [MF25].

The present thesis is primarily concerned with the operational level; however, it also considers the moment before execution, when elective patients are selected and scheduled, and the one that arises during schedule execution, when uncertainty has been partially revealed, and the initially planned schedule may need to be adapted. We can get surgical durations substantially different from their expected values, unexpected emergency or non-elective patient arrivals, and delays propagating through sequences of surgeries. As a consequence, a schedule that appears feasible when constructed may become difficult to complete within regular operating hours. Recent reviews of operating room scheduling identify duration variability, patient uncertainty, resource limitations, and staff availability among the main factors that make deterministic schedules difficult to implement without adjustment [ABK25].

These characteristics motivate the use of stochastic and adaptive approaches rather than relying exclusively on schedules constructed from expected durations.

2.2 Real-Time, Cancellation and Non-Elective Demand

When an operating room session deviates from its original plan, one of the main operational decisions concerns the trade-off between extending activity into overtime and postponing elective surgery. Performing the next planned case preserves surgical throughput and avoids postponement for the patient, but may increase staffing costs and workload. Cancellation or postponement of the case protects capacity and limits overtime, but transfers the consequences of uncertainty to the patient and to the surgical waiting list.

The appropriate decision therefore depends on the current state of the system rather than only on the schedule originally constructed. Relevant information includes elapsed time, remaining capacity, the expected workload of the surgeries still planned, and any high-priority demand that has already become known.

Duma and Aringhieri [DA18] study this type of real-time operating room management problem and show the relevance of adaptive policies that react to the actual evolution of the operating session. Their work is particularly relevant to the decision considered in this thesis, since overtime and postponement decisions are taken sequentially as information becomes available.

The presence of non-elective patients makes the problem more complex. Unlike elective patients, their arrival times are not known when the original schedule is constructed, and some must be treated within relatively short time limits. The trade-off between integrating non-elective demand into shared operating rooms and reserving dedicated emergency capacity has been extensively studied [VD15; Wul+07]. Duma and Aringhieri [DA19] analyze policies for the joint management of elective and non-elective patients and distinguish between dedicated and shared operating room configurations.

Dedicated capacity limits interference with the elective schedule but can leave resources unused, whereas shared capacity increases flexibility at the cost of stronger interaction between elective and urgent activity.

The framework developed in this thesis adopts the shared-capacity perspective. This means that non-elective patients compete with elective surgeries for available operating room time. It is possible to use complete stochastic realizations in an ex-post optimization model in order to create benchmark decisions. On the other hand, an implementable online policy has to base its decisions only on the arrivals and workload which have already become observable.

The distinction between perfect-information benchmarks and implementable real-time policies forms the basis for the later discussion of the optimization and learning components.

2.3 Optimization-Based Approaches under Uncertainty

Mathematical programming has played a central role in operating room planning and scheduling. Mixed-Integer Linear Programming formulations are particularly suitable because many relevant decisions–selecting a patient, assigning a surgery to a room, opening capacity, authorizing overtime, or canceling a procedure–have a discrete structure [BD07; CDB10].

Within operating room scheduling, optimization models have been proposed for both elective and non-elective activity and under patient-centered, cost-oriented, and resource-utilization objectives [DA19; Bel+24]. More recent surveys continue to identify mathematical programming as one of the main methodological families in the field and emphasize the increasing attention given to uncertainty and realistic operational constraints [ABK25].

When uncertain quantities are represented probabilistically, stochastic programming provides a natural extension of deterministic scheduling. Decisions taken before uncertainty is observed can be distinguished from recourse decisions which adapt to realized scenarios. In practice, the underlying distributions often lead to stochastic programs whose expectations cannot be evaluated exactly. Scenario-based approximations, including Sample Average Approximation, thus provide a practical way of replacing the full uncertainty distribution with a finite collection of Monte Carlo realizations.

This approach is notably appropriate when uncertainty in surgical durations and non-elective workload must be considered already when constructing the elective schedule. At the same time, a large number of scenarios can make mixed-integer stochastic programs computationally demanding, motivating scenario-reduction techniques that retain a smaller weighted set of representative realizations.

Recent stochastic optimization models explicitly combine surgery-duration uncertainty with stochastic emergency demand in ex-ante operating-room scheduling [Gon+25]. Related model-driven research also addresses patient admission and surgical-case scheduling under uncertainty through mathematical programming and scenario-based solution methods [Liu24].

Optimization also provides an important benchmark after uncertainty has been realized. If all realized durations and arrivals are assumed to be known, an ex-post model can determine which decisions would have been optimal with perfect information. Such a solution is not directly implementable because future information is unavailable during actual execution, but it can be used both as a performance benchmark and as a source of high-quality decisions from which a data-driven policy can learn.

The main limitation is computational. Solving a detailed mixed-integer optimization problem once offline may be entirely feasible, whereas repeatedly solving it whenever a new decision is required can be impractical in a real-time environment. This tension between optimization quality and online computational speed creates a natural motivation for combining mathematical optimization with machine learning.

2.4 Learning from Optimization for Online Decision Support

The distinction between offline and online decision making is especially important when decisions are sequential and irreversible. An offline optimization model can exploit the complete realization of uncertain quantities, whereas an online policy has access only to the information revealed up to the current decision time. For this reason, we can interpret the corresponding offline solution as a perfect-information oracle since it provides an ideal reference but cannot be implemented directly.

An approach to leveraging such an oracle is to use its decisions as training examples: instead of repeatedly solving the optimization problem online, a supervised model learns a mapping from the currently observable state to the action selected by the optimizer. Pham et al. [Pha+23], for example, use offline optimization solutions to learn scheduling decisions in a radiotherapy context, demonstrating how optimization-generated decisions can be transferred to a predictive policy.

This idea is related to a larger and rapidly growing literature combining machine learning and mathematical optimization. Recent surveys distinguish several paradigms, including predict-then-optimize, contextual optimization, and decision-focused learning. In predict-then-optimize approaches, machine learning typically estimates uncertain parameters that are subsequently passed to an optimization model. For instance, Daldossi et al. use a predict-then-optimize framework for selecting and sequencing surgeries in an operating theater, showing that a better precision in prediction does not produce better decisions [Dal+26]. Hence, decision-focused learning goes a step further by training predictive models according to the quality of the downstream decisions rather than prediction error alone [Man+24].

This distinction is relevant because predictive quality and operational decision quality are not necessarily equivalent. A classifier or regression model may achieve better conventional predictive metrics without producing better downstream decisions. The decision-focused learning literature explicitly emphasizes this possible misalignment between predictive losses and the objective ultimately relevant to the decision maker [Man+24].

The approach developed in this thesis is conceptually related to this literature, but

it is not an end-to-end decision-focused learning method in the strict sense. The classifier is trained through standard supervised learning on decisions generated by a full-information oracle. Operational information enters later in model and threshold selection, where predictive metrics are considered together with the quality of the resulting operating room decisions.

2.5 Machine Learning in Operating Room Management

Machine learning has become increasingly common in perioperative management as richer clinical and operational datasets have become available. A recent systematic review by Bellini et al. examines 22 studies published between 2019 and 2023 and identifies surgery-duration prediction, resource allocation, and cancellation detection among the principal applications of artificial intelligence in operating room management. The review also reports the use of methods such as Random Forests and XGBoost in this context [Bel+24].

These applications are primarily predictive: the model estimates an uncertain quantity or event that may subsequently support planning decisions. A recent example is provided by Lex et al., who combine patient-specific machine-learning predictions of surgery duration with optimization models for elective orthopedic scheduling. Their predict-then-optimize framework cuts overtime relative to schedules based on average durations, while also illustrating the trade-off between overtime and underutilization [Lex+25].

The framework considered in this thesis follows a different direction. Instead of using machine learning primarily to estimate an uncertain input to a subsequent optimization model, the learning task directly approximates an operational execute/cancel decision previously generated by an optimization oracle. This shifts the prediction target from a system parameter to the decision itself.

Several classifier families can be used for this purpose. Simpler models such as Logistic Regression and Decision Trees offer comparatively transparent decision structures, whereas ensemble methods such as Random Forests and gradient-boosted trees can capture more complex nonlinear interactions between state variables. In operating room applications, where elapsed time, residual capacity, remaining workload, patient characteristics, and non-elective demand interact, comparing models with different degrees of flexibility is therefore preferable to assuming in advance that a single model family will be most appropriate.

Moreover, class imbalance is a factor when the number of cancellations is much smaller than the number of executions. In this case, accuracy alone gives an incomplete picture of the model's performance. Information is provided by metrics focused on the minority class, for example, the F1-score for cancellations, and by threshold-independent measures such as ROC-AUC. Most importantly, for the purpose of supporting decisions, these predictive metrics should be interpreted in conjunction with the operational consequences of the actions that are predicted.

2.6 Discrete-Event Simulation for Dynamic Policy Evaluation

Predictive evaluation on a fixed dataset is not sufficient to characterize a policy that will be deployed sequentially. Once a decision is implemented, it changes the future state of the system: executing a surgery consumes time and capacity, while canceling it changes the remaining workload. Non-elective arrivals and room availability additionally affect which states will subsequently be reached. The distribution of decision states encountered under a learned policy may consequently differ from the distribution observed in the offline dataset.

Discrete-Event Simulation (DES) is well suited to this type of evaluation because it explicitly represents the chronological evolution of a system through events such as patient arrivals, surgery starts and completions, turnover operations, and operating room releases. DES has a long history in healthcare operations, where it has been used to study patient flows, resource capacity, waiting times, and scheduling policies [JJS99]. More recent reviews confirm that healthcare DES is increasingly used in combination with optimization and other analytical methods, particularly for operational decision support[FB22].

Recent applications also demonstrate its relevance specifically to operating room management. Pu et al. use a discrete-event model to investigate operating room efficiency across different surgical specialties, illustrating the ability of simulation to study throughput under stochastic workflow conditions [PWH24].

An especially relevant recent contribution is the 2026 study by Çalışkan et al., which compares rule-based, predict-then-optimize, genetic-algorithm, and reinforcement-learning policies within a common discrete-event simulation environment. The policies are evaluated under identical stochastic conditions and shared resource constraints using common random numbers across 500 replications. The authors emphasize DES as a controlled environment for comparing online scheduling strategies before real-world implementation [Çal+26].

This type of evaluation is closely related to the objective of the final phase of this thesis. The classifier is not evaluated only on independently stored decision states: it is embedded in an event-driven simulator, queried at the states actually reached by its own decisions, and compared with alternative policies under common stochastic realizations. The final experiment in the thesis uses 1,000 replications for each relevant instance–specialty combination, with the expected number of simulation records obtained and common-random-number checks successfully verified.

2.7 Research Gap and Thesis Positioning

The literature reviewed above shows that operating room management under uncertainty has been addressed through a broad range of optimization, simulation, and progressively data-driven approaches. Mathematical programming provides detailed representations of resource constraints and competing objectives, but its computational requirements may

limit repeated use during real-time execution. Machine learning provides rapid inference, but recent reviews show that its role in operating room management has primarily concerned predictive tasks such as estimating surgery durations, resource requirements, or cancellation risks [Bel+24].

Recent predict-then-optimize approaches have begun to connect these two areas by passing machine-learning predictions to downstream scheduling models [Lex+25], while the wider decision-focused learning literature stresses that predictive performance and downstream decision quality need not coincide [Man+24]. Nevertheless, comparatively less attention has been devoted to learning a real-time execute/cancel policy directly from perfect-information optimization decisions, particularly when elective and stochastic non-elective demand interact using shared operating room capacity.

The thesis describes a stochastic scheduling model which begins by preparing elective schedules on the basis of patient-related and facility-related objectives. Surgical duration uncertainty and non-elective demand are dealt with by means of full case-mix scenario generation and scenario reduction. A retroactive MILP oracle is then used to determine the execute/cancel decisions that would be optimal if all the information were available; these decisions are subsequently turned into supervised-learning labels. As the oracle might make use of information from the future to set the target, but the actual future outcomes are never given to the online classifier, we also base the chronological reconstruction of the input states associated with those labels only on information that would have been available at the time each decision was made. The candidate classifiers and decision thresholds are then assessed using both predictive and operational criteria.

Finally, the selected policy is embedded in an event-driven discrete-event simulation in which it interacts with stochastic surgical durations and chronologically revealed non-elective arrivals. The closed-loop evaluation makes it possible to assess not only agreement with the offline oracle but also the operational consequences of the learned policy and the state trajectories induced by its own decisions.

In this way, the contribution of the thesis lies in the connection of four components that are often studied separately: stochastic operating room scheduling, perfect-information optimization, supervised policy approximation, and dynamic simulation-based evaluation. We do not aim to replace optimization with machine learning, but to use optimization offline to transfer high-quality decision knowledge to a computationally inexpensive online policy.

Chapter 3

Problem Statement and Offline Optimization Models

In this chapter, we define the main operational problem addressed in this thesis, namely the real-time allocation of operating room overtime in an uncertain environment. We present an optimization framework that consists of two separate offline problems. In the first problem, an ex-ante weekly scheduling model chooses the elective patients and allocates the initial room capacity before the actual durations are known. Starting from a waiting list, we use this model to obtain an optimized weekly surgery schedule, which consists of an instance of the online operating-room problem. In the second problem, an ex-post optimization oracle assesses the schedule once all the scenarios have been realized and then determines the best execute-or-cancel decisions retroactively. The aim of this second model is to provide a dataset of (ex post) optimized instances of the online operating-room problem, which can be used to learn a policy for the real-time overtime allocation.

3.1 Online Overtime Allocation Problem

As an operating session progresses, the remaining elective workload may cease to fit within the available regular time. When capacity is insufficient, it is unavoidable to decide whether to perform the next scheduled procedure or postpone the intervention. Carrying out the procedure keeps elective activities intact and prevents postponement, but uses up residual capacity and could result in overtime; on the other hand, canceling or postponing the procedure safeguards the remaining capacity but shifts the consequences to the patient and the future schedule. The right course of action cannot be established just from the original schedule since it depends on the current state of the system, such as the time that has elapsed since the session began, the residual capacity, the remaining elective workload, and the non-elective demand that has already been observed.

Table 3.1: Decision levels in the proposed framework.

Decision level	Timing	Information	Decision
Ex-ante weekly model	Before realization	Stochastic scenarios	Patient selection and OR assignment
Ex-post oracle	After realization	Complete scenario	Execute/cancel on fixed schedule
Online policy	During execution	Observed state	Execute/cancel next elective case

3.2 Ex-Ante Weekly Scheduling Model

The initial elective schedules are generated through a stochastic scheduling model with first-stage assignment decisions and scenario-dependent recourse decisions. The model is implemented through a two-phase optimization procedure. The first phase solves a deterministic assignment problem based on expected clinical durations, while the second phase augments the model with sampled duration scenarios and optimizes the corresponding recourse criterion.

In this optimization framework, patient positions are *not* decision variables of the stochastic MILP. The optimization determines which patients are scheduled and their operating room assignment. Once a candidate assignment has been obtained, the patients assigned to each operating room are ordered by decreasing priority, with shorter expected clinical duration used to break priority ties. This ordered representation is subsequently used in all the other stages. The complete formulation of the two-phase optimization is reported below.

Let $\mathcal{I}$ denote the set of elective patients and $\mathcal{R}$ the set of available operating rooms. For each patient $i \in \mathcal{I}$, let $\bar{d}_i$ be the expected clinical duration and q_i the synthetic priority score. The regular session length is denoted by C, while $\bar{O}$ represents the maximum overtime allowed for each operating room. The values used in the computational experiments are reported in Chapter 7.

A subset $\mathcal{I}^M \subseteq \mathcal{I}$ of high-priority patients must be scheduled. In Chapter 7, we report the rule used to define this mandatory set in the computational experiments.

The model consists of first-stage scheduling decisions and scenario-dependent recourse decisions.

First-stage scheduling decisions. The binary variables are:

$$x_{ir} = \begin{cases} 1 & \text{if patient } i \text{ is assigned to operating room } r, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$u_i = \begin{cases} 1 & \text{if patient } i \text{ is included in the schedule,} \\ 0 & \text{otherwise.} \end{cases}$$

Each scheduled patient is assigned to exactly one operating room:

$$\sum_{r\in\mathcal{R}} x_{ir} = u_i, \qquad \forall i \in \mathcal{I}.$$

Patients belonging to the set $\mathcal{I}^M$ are mandatory:

$$u_i = 1, \qquad \forall i \in \mathcal{I}^M.$$

At this stage, capacity is evaluated using expected clinical durations. For each operating room,

$$\sum_{i\in\mathcal{I}} \bar{d}_i x_{ir} \le C + \bar{O}, \qquad \forall r \in \mathcal{R}.$$

Turnover is not included in this initial assignment constraint; the first-phase capacity calculation is based on expected clinical duration only.

The primary objective depends on the scheduling strategy. Under the patient-centered strategy:

$$\max Z_1^P = \sum_{i\in\mathcal{I}} q_i u_i,$$

whereas under the facility-centered strategy:

$$\max Z_1^F = \sum_{i\in\mathcal{I}} \bar{d}_i u_i.$$

Scenario-dependent recourse decisions. Let Ω be the set of duration scenarios. For each scenario $\omega \in \Omega$, let p_ω be its probability, with

$$p_\omega \ge 0, \qquad \sum_{\omega\in\Omega} p_\omega = 1,$$

and let $\widetilde{d}_i^{\omega}$ denote the realized clinical duration of patient i under scenario ω.

For each scenario, the binary recourse variable

$$v_{ir}^{\omega} = \begin{cases} 1 & \text{if patient } i \text{ is served in room } r \text{ under scenario } \omega, \\ 0 & \text{otherwise,} \end{cases}$$

and the cancellation variable

$$c_i^{\omega} = \begin{cases} 1 & \text{if scheduled patient } i \text{ is canceled under scenario } \omega, \\ 0 & \text{otherwise,} \end{cases}$$

are introduced. In addition,

$$O_r^{\omega} \ge 0$$

denotes the overtime used by operating room r in scenario ω.

A patient can be served in an operating room only if assigned to that room:

$$v_{ir}^{\omega} \leq x_{ir}, \qquad \forall i \in \mathcal{I},\ r \in \mathcal{R},\ \omega \in \Omega.$$

Every scheduled patient is either served or canceled:

$$\sum_{r \in \mathcal{R}} v_{ir}^{\omega} + c_i^{\omega} = u_i, \qquad \forall i \in \mathcal{I},\ \omega \in \Omega.$$

For each operating room and scenario, the realized clinical workload must satisfy

$$\sum_{i \in \mathcal{I}} \widetilde{d}_i^{\omega} v_{ir}^{\omega} \leq C + O_r^{\omega}, \qquad \forall r \in \mathcal{R},\ \omega \in \Omega,$$

with

$$0 \leq O_r^{\omega} \leq \bar{O}.$$

The secondary objective again depends on the selected managerial strategy. For the patient-centered strategy, the expected number of cancellations is minimized:

$$\min Z_2^P = \sum_{\omega \in \Omega} p_{\omega} \sum_{i \in \mathcal{I}} c_i^{\omega}.$$

For the facility-centered strategy, the expected overtime is minimized:

$$\min Z_2^F = \sum_{\omega \in \Omega} p_{\omega} \sum_{r \in \mathcal{R}} O_r^{\omega}.$$

Under the patient-centered strategy, Z_1^P is maximized first and Z_2^P is subsequently minimized among solutions attaining the optimal primary-objective value. Under the facility-centered strategy, instead, Z_1^F is maximized first and Z_2^F is subsequently minimized among solutions attaining the optimal primary-objective value.

The variable domains are

$$x_{ir}, u_i, v_{ir}^{\omega}, c_i^{\omega} \in \{0, 1\},$$

and

$$O_r^{\omega} \geq 0.$$

Construction of the operating room sequence. Patient selection and operating room assignment are determined by the stochastic MILP without positional decision variables. Once we obtain a candidate solution, we order patients assigned to each operating room in a deterministic manner.

Patients are initially sorted so that if two patients have the same priority, the patient with the shorter expected clinical duration is prioritized. The patient identifier then serves as a final deterministic tie-breaker. The sequence we get is fixed in the later scenario-reduction, oracle, and chronological replay stages. This separation keeps

the initial stochastic optimization substantially smaller while still producing a complete ordered schedule for the subsequent parts of the framework.

3.3 Ex-Post Perfect-Information Oracle

Once we obtain a weekly schedule, we keep its elective patient selection, operating room assignment, and within-room order fixed. Then, for each scenario, the model determines which scheduled elective surgeries should be executed and which should be canceled.

This decision is produced by a deterministic mixed-integer optimization model that operates with complete knowledge of the scenario under consideration.We thus use the model serves as a perfect-information benchmark rather than an implementable online policy. Its primary function is to generate high-quality execute or cancel decisions that can subsequently be used as target labels for supervised learning.

Unlike the stochastic scheduling model of Section 3.2, the oracle does not reconsider the elective scheduling task. Patient selection and operating room assignment have already been determined, and the within-room sequence has already been constructed. The oracle only decides how far each elective sequence should be executed once the scenario has been revealed.

Fixed elective sequences. Let $\mathcal{J}$ denote the set of operating rooms contained in the weekly schedule. For each room $j \in \mathcal{J}$, let

$$P_j = \left(i_{j1}, i_{j2}, \ldots, i_{jm_j}\right)$$

be its ordered sequence of elective surgeries, where m_j is the number of elective patients assigned to room j.

For each scheduled elective patient i, let d_i^s denote the realized clinical duration in scenario s, τ_i the corresponding turnover time, and q_i the patient priority score.

The binary decision variable

$$x_i^s = \begin{cases} 1, & \text{if elective patient } i \text{ is executed in scenario } s, \\ 0, & \text{if elective patient } i \text{ is canceled,} \end{cases}$$

defines the oracle decision that will later become the supervised-learning label.

Since cancellations are assumed to occur only at the end of an operating room sequence, the following prefix constraints are imposed:

$$x_{i_{jk}}^s \geq x_{i_{j,k+1}}^s, \qquad j \in \mathcal{J}, \quad k = 1, \ldots, m_j - 1.$$

Hence, if an elective surgery is canceled, all subsequent elective surgeries in the same room are also canceled.

Non-elective workload. Let $\mathcal{Q}^s$ denote the set of non-elective patients generated in scenario s. Each non-elective patient $r \in \mathcal{Q}^s$ has a day of arrival a_r and a realized

operating room workload

$$\ell_r^s = d_r^s + \tau_r,$$

including its turnover component.

Let $\mathcal{J}(a_r)$ denote the operating rooms associated with the same day as patient r. The binary variable

$$y_{rj}^s = \begin{cases} 1 & \text{if non-elective patient } r \text{ is assigned to room } j, \\ 0 & \text{otherwise,} \end{cases}$$

is defined only for $j \in \mathcal{J}(a_r)$.

A binary slack variable u_r^s is also introduced to identify a non-elective patient that cannot be accommodated:

$$\sum_{j \in \mathcal{J}(a_r)} y_{rj}^s + u_r^s = 1, \qquad r \in \mathcal{Q}^s.$$

The variable u_r^s is heavily penalized and is included only to avoid model infeasibility in extreme scenarios.

In the oracle, we represent non-elective demand at the daily capacity level. Arrival times and response-time limits are known as part of the complete scenario, but the oracle does not construct a chronological within-day sequence of non-elective arrivals. Their event-by-event interaction with the elective schedule is introduced in the chronological replay of Section 5.1.

Capacity and overtime. Let H denote the length of the regular operating session and let B denote the maximum regular overtime available to each room. The values used in the computational experiments are reported in Chapter 7.

Let

$$OT_j^s \geq 0$$

denote regular overtime and

$$EOT_j^s \geq 0$$

an additional extra-overtime variable. We introduce the latter only to preserve feasibility when the realized non-elective demand cannot be accommodated within regular capacity and the standard overtime allowance.

For operating room j, total realized workload satisfies

$$\sum_{k=1}^{m_j} d_{i_{jk}}^s x_{i_{jk}}^s + \sum_{k=1}^{m_j-1} \tau_{i_{jk}} x_{i_{j,k+1}}^s + \sum_{\substack{r \in \mathcal{Q}^s \\ j \in \mathcal{J}(a_r)}} \ell_r^s y_{rj}^s \leq H + OT_j^s + EOT_j^s, \qquad j \in \mathcal{J}.$$

The turnover following elective patient i_{jk} is counted only when the next elective surgery

in the sequence is executed. Furthermore,

$$0 \leq OT_j^s \leq B.$$

The extra-overtime variable is not subject to the standard 30-minute bound. Instead, its use is minimized before the elective objective is considered.

Lexicographic solution procedure. The first optimization stage of the oracle is independent of the elective scheduling strategy and gives priority to the feasibility of non-elective demand.

Let $\zeta_{\mathrm{NE}} > 0$ denote a sufficiently large penalty coefficient associated with unserved non-elective patients, and let $\varepsilon_F > 0$ denote the numerical tolerance we used to retain the optimal value of the optimal value of the first oracle stage. Its goal is to minimize

$$F^s = \sum_{j \in \mathcal{J}} EOT_j^s + \zeta_{\mathrm{NE}} \sum_{r \in \mathcal{Q}^s} u_r^s.$$

Let $F^{s,*}$ denote the minimum value. We restricted the subsequent optimization stages thus to solutions satisfying

$$F^s \leq F^{s,*} + \varepsilon_F.$$

We adopt this ordering because it strongly discourages leaving non-elective patients unserved and, conditional on serving them, minimizes the additional overtime required to preserve feasibility. The remaining objective depends on the scheduling strategy.

For the patient-centered strategy, let $\varepsilon_P > 0$ denote the tolerance we used to retain the optimal patient-priority value, and let $\eta_P > 0$ denote a sufficiently small tie-breaking coefficient. The primary objective is

$$\max P^s = \sum_i q_i x_i^s.$$

Let $P^{s,*}$ denote its best value. The secondary optimization retains the primary objective within the tolerance

$$P^s \geq P^{s,*} - \varepsilon_P,$$

and minimizes

$$\sum_i (1 - x_i^s) + \eta_P \sum_{j \in \mathcal{J}} OT_j^s.$$

Thus, after preserving the priority of executed patients, the patient-centered oracle favors fewer elective cancellations, with regular overtime used only as a small tie-breaking term.

For the facility-centered strategy, let $\varepsilon_U > 0$ denote the tolerance used to retain the optimal elective-workload value, and let $\eta_U > 0$ denote a sufficiently small tie-breaking coefficient. The primary objective maximizes the realized elective clinical workload,

$$\max U^s = \sum_i d_i^s x_i^s.$$

After obtaining $U^{s,*}$, the model imposes

$$U^s \geq U^{s,*} - \varepsilon_U,$$

and minimizes

$$\sum_{j \in \mathcal{J}} OT_j^s + \eta_U \sum_i (1 - x_i^s).$$

The facility-centered oracle therefore first preserves realized elective activity and then reduces regular overtime, using the number of cancellations only as a tie-breaking term.

Oracle labels. The final oracle decision is stored for every scheduled elective patient as

$$L_i^s = \begin{cases} 1 & \text{if } x_i^s = 1, \\ 0 & \text{if } x_i^s = 0. \end{cases}$$

The target variable accordingly is not an overtime decision, but an execute/cancel decision; in fact, the classifier does not predict overtime, despite the feasibility and objective structure that produce the label depending on it.

Chapter 4

Solution Method for the Weekly Scheduling Problem

4.1 Sample Average Approximation with Multiple Replications

The stochastic scheduling model depends on uncertain surgical durations of the elective procedures. Let ξ denote the random vector of these durations and let z represent the common first-stage scheduling decisions. For a fixed schedule, its expected recourse performance can be written abstractly as

$$\mathbb{E}_{\xi}[Q(z,\xi)]\,.$$

Since we cannot evaluate this expectation analytically for the empirical duration distributions used in the study, we approximate it through Monte Carlo sampling. Given a finite sample

$$\Omega_N = \{1,\ldots,N\},$$

the expectation is replaced by

$$\widehat{Q}_N(z) = \frac{1}{N}\sum_{\omega\in\Omega_N} Q(z,\xi^{\omega}).$$

Due to the complexity of the resulting mixed integer programming model, rather than solving a single stochastic program on the complete Monte Carlo sample, we use multiple SAA replications. The scenario indices are randomly shuffled and divided into non-overlapping folds, and a stochastic scheduling problem is solved independently on each fold, producing alternative candidate schedules.

Each fold-specific stochastic problem is solved through a two-phase computational procedure. In the first phase, the deterministic first-stage problem is solved using expected surgical durations.

Let $\widehat{u}_i$ and $\widehat{x}_{ir}$ represent the incumbent solution obtained within the available com-

putational budget, and let Z_1^{inc} represent its primary-objective value. Because this phase is subject to a time limit and a positive MIP gap tolerance, Z_1^{inc} is the value of the best available incumbent and is not necessarily the global optimum.

In the second phase, the stochastic problem is solved over the scenario fold Ω_k. Patients selected by the first-phase incumbent solution are retained by imposing

$$u_i \geq \widehat{u}_i, \qquad i \in \mathcal{I},$$

while the first-stage primary-objective value is preserved through

$$Z_1 \geq Z_1^{\text{inc}}.$$

The operating room assignments $\widehat{x}_{ir}$ are supplied as a warm start but are not fixed. Accordingly, the assignment variables x_{ir} may change subject to the original assignment and capacity constraints. Within each SAA fold, all scenarios receive equal probability:

$$p_\omega = \frac{1}{|\Omega_k|}, \qquad \omega \in \Omega_k.$$

The candidate obtained from each fold is subsequently evaluated on the common Monte Carlo scenario set. The final candidate is selected based on the managerial objective corresponding to the scheduling strategy. In the patient-centered strategy, higher realized priority and lower cancellation rates are prioritized; while in the facility-centered strategy, higher utilization and reduced overtime are prioritized. Additional performance indicators serve as tie-breakers only when necessary.

This full-sample comparison is an important part of the computational design. The replication procedure is thus not used as conventional cross-validation for statistical model assessment. We use it to generate several stochastic candidate schedules from different subsets of the Monte Carlo sample and select the candidate that performs best when evaluated on the common scenario pool.

4.2 Full Case-Mix Scenario Reduction and Reduced SAA Schedule Generation

The initial SAA procedure described in Section 4.1 provides one elective schedule for each case-mix instance and scheduling strategy. A second, larger Monte Carlo sample is then generated to obtain a more detailed representation of surgical-duration uncertainty and to construct the reduced stochastic problems used in the remainder of the pipeline.

A key feature of this stage is that scenario reduction is performed on the *complete case mix*, rather than only on the patients selected in the initial SAA_N schedule. This allows the subsequent reduced-SAA optimization to reconsider every patient available in the original case mix, including those excluded from the preliminary schedule.

Elective scenario representation and clustering. We generate a larger set of independent Monte Carlo realizations of elective clinical durations from the patient-specific distributions. Let M denote the number of scenarios in this larger sample.

For a case mix that contains n patients, scenario s is represented by the vector

$$\mathbf{d}^s = (d_1^s, d_2^s, \ldots, d_n^s)\,,$$

where d_i^s denotes the realized clinical duration of patient i in scenario s.

In turn, the scenario matrix is defined as

$$X = \begin{bmatrix} d_1^1 & d_2^1 & \cdots & d_n^1 \\ d_1^2 & d_2^2 & \cdots & d_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ d_1^M & d_2^M & \cdots & d_n^M \end{bmatrix}.$$

Each row of X represents one realization of the complete patient population, while each column corresponds to the same patient across the M scenarios. In this way, patients not included in the initial SAA_N schedule still contribute to the distance between scenarios and to the selection of representative realizations.

K-means clustering is applied to the scenario matrix. The experimental choices concerning standardization, the number of clusters, and the number of initializations are reported in Chapter 7.

Let $\mathcal{C}_h$ denote cluster h and let

$$\boldsymbol{\mu}_h = \frac{1}{|\mathcal{C}_h|} \sum_{s \in \mathcal{C}_h} \mathbf{d}^s$$

be its centroid. Since the centroid itself does not generally correspond to an observed Monte Carlo realization, one actual scenario is selected from each cluster. The representative scenario is the cluster member with the smallest Euclidean distance from the centroid:

$$s_h^\star = \arg\min_{s \in \mathcal{C}_h} \|\mathbf{d}^s - \boldsymbol{\mu}_h\|_2\,.$$

The term *representative scenario* is used throughout the thesis for this nearest-to-centroid realization. This distinction is useful because the procedure is based on K-means rather than on a K-medoids clustering algorithm.

Each representative scenario receives the empirical probability of the corresponding cluster,

$$\pi_h = \frac{|\mathcal{C}_h|}{M}, \qquad h = 1, \ldots, K,$$

so that

$$\sum_{h=1}^{K} \pi_h = 1.$$

The reduced scenario set therefore contains K observed duration realizations together

with their empirical weights.

Scenario extension with non-elective patients. The extended scheduling model additionally includes stochastic non-elective demand. Non-elective patients are not part of the elective case mix and are not first-stage scheduling decisions. Instead, they represent external demand for the same operating room capacity.

For each scenario s and each day d, the number of non-elective arrivals is sampled as

$$A_d^s \sim \text{Poisson}(\lambda).$$

The parameter λ is derived from a target non-elective share q of total expected surgical demand. If n_{ref} denotes the common elective reference for a given instance–specialty pair and D is the number of days in the planning horizon, then

$$\lambda = \frac{1}{|D|} \frac{q}{1-q} n_{\text{ref}}.$$

Arrival times are sampled within the regular operating session. Surgical durations are generated by sampling a template from the corresponding elective case-mix and applying the same patient-duration generation mechanism used for elective procedures. The non-elective workload also includes the associated turnover component.

Non-elective uncertainty is incorporated into scenario reduction by summarizing the event list using day-level features while maintaining a fixed-dimensional representation. For each day in the planning horizon, the clustering vector includes the number of non-elective arrivals, realized and expected non-elective workload, the number of trauma, emergent, and urgent arrivals, as well as the minimum response-time limit among the arrivals for that day.

We also concatenated these variables with the complete elective-duration vector prior to the application of K-means clustering. Both elective duration uncertainty and the level of capacity pressure generated by non-elective demand are thus summarized by representative scenarios in the final non-elective experiments.

Weighted reduced SAA and out-of-sample evaluation. The representative scenarios are used to solve a new stochastic scheduling problem over the full elective case mix. Unlike the initial SAA problems, where the scenarios inside each fold receive equal weight, the reduced formulation uses the empirical cluster probabilities.

For a first-stage solution z, the expected recourse term is approximated by

$$\mathbb{E}\left[Q(z,\xi)\right] \approx \sum_{h=1}^{K} \pi_h Q\left(z, \xi^{s_h^\star}\right).$$

Here, $\xi^{s_h^\star}$ denotes the duration realization associated with the representative scenario selected from cluster h. The reduced-SAA schedule is therefore the solution of a new weighted stochastic optimization problem.

When non-elective demand is included, the elective capacity available in the first-stage assignment model is reduced by the expected representative non-elective workload of each day, and the reserved workload is distributed across the operating rooms associated with that day. In each representative scenario, in addition, the realized non-elective daily workload is added to the corresponding recourse capacity constraints, again distributed across the operating rooms of the day.

We interpret this treatment as an aggregate approximation of a shared operating room policy. At this stage, non-elective patients are represented through their effect on available capacity; their chronological insertion into individual operating rooms is not simulated. The event-by-event dynamics are introduced later, when the online decision states are reconstructed in Section 5.1 and when the final policy is evaluated through discrete-event simulation.

Before the reduced-SAA schedules are used for oracle label generation, their elective-only version is evaluated on the original larger Monte Carlo sample. For the out-of-sample comparison, the elective patient set selected by each candidate solution is fixed, while operating room assignments are allowed to adapt to each scenario. Thus, this experiment assesses the robustness of the initial decision regarding patient selection rather than the performance of the entire fixed operating room schedule. This provides an out-of-sample comparison of the two first-stage solutions under a common set of realizations.

The results of this comparison are reported in Section 8.1.

Chapter 5

Optimization-Guided Learning of the Online Policy

5.1 Chronological Reconstruction of Online Decision States

The perfect-information oracle determines which elective surgeries should be executed in each complete scenario, but the state variables used by an online policy must satisfy a different information requirement. At a real decision time, future non-elective arrivals and future realized durations are not yet known.

Future-information leakage would be introduced if we used the oracle information set directly as input to the classifier: when an actual online decision must be made, the resulting model would be trained on quantities that are unavailable. We therefore use the oracle only to define the target action; while the execute/cancel label is preserved, the predictor vector from information observable up to the corresponding decision time is reconstructed by a separate event-driven replay.

We summarize this distinction as

$$\underbrace{L_i^s}_{\text{target}} \leftarrow \text{perfect-information oracle}, \qquad \underbrace{\mathbf{x}_i^s}_{\text{predictors}} \leftarrow \text{chronological replay}.$$

The chronological reconstruction does not solve the oracle optimization problem again and does not alter its labels. We use it only to associate each label with a state based on information that could have been available when the corresponding decision was reached.

Event-driven replay and reachability. The replay is performed separately for each day of the weekly schedule. At the beginning of the day, each operating room is available at time $t = 0$, where time is measured in minutes from the beginning of the regular session. Thus, for a session starting at 08:00, the numerical value $t = 0$ corresponds to 08:00.

The clock in the simulation moves on from one event to the next. The relevant events are non-elective arrivals and operating room release times.

Only non-elective patients who satisfy the following condition

$$a_r \leq t$$

are revealed at time t, while future arrivals remain unobserved in the current system state, although their simulated event times are internally maintained by the replay engine.

When one or more observed non-elective patients are waiting, and an operating room becomes available, non-elective demand is prioritized over elective surgeries. Waiting patients are sequenced first by due time and then, in the case of ties, by arrival time and identifier. The considered operational rule is that a non-elective arrival that occurs during an ongoing surgery must wait until an operating room becomes available.

If non-elective patient r begins service at time t, the corresponding operating room is released at

$$t' = t + d_r^s + \tau_r.$$

When no observed non-elective patient is waiting, an idle operating room proceeds to the decision point for its next scheduled elective surgery.

The state is recorded *before* the elective decision is applied. If the oracle label is one, the elective surgery is executed and the operating room clock advances by its realized duration and, except after the last planned elective surgery in the room, its turnover time.

If the oracle label is zero, this is the first reachable cancellation in that operating room. Because the oracle labels satisfy the prefix property, all subsequent elective surgeries in the same room also have label zero, but those later rows are not additional decisions that would actually be reached after the cancellation.

More formally, let

$$L_{jk}^s \in \{0, 1\}$$

be the oracle label of the surgery in position k of room j. If a cancellation occurs, define

$$k_j^0 = \min \left\{ k : L_{jk}^s = 0 \right\}.$$

Then the states retained as reachable are

$$k \leq k_j^0.$$

In particular, all executed surgeries preceding k_j^0 and the first cancellation itself are genuine decision points, whereas positions

$$k > k_j^0$$

belong to the canceled suffix and are marked as unreachable. If no cancellation occurs in the operating room block, all scheduled elective decision points remain reachable.

Two versions of the reconstructed dataset are stored. The *all-row* version retains one row for every scheduled elective surgery and is used for diagnostic checks. The *reachable* version removes prefix-induced suffix rows and is the version used for supervised learning.

This construction is oracle-guided because the oracle labels determine which elective surgeries are executed and, in turn, which states are visited. It should not be interpreted as a second optimization model. The learned policy may visit a different sequence of states when deployed; this is one of the reasons why predictive evaluation is complemented by the dynamic simulation experiments of Chapter 6.

Online-safe state variables. At each reachable elective decision point, the replay constructs a state vector containing only variables that are available from the schedule or from events already observed by the current time. The final implementation uses 27 online-safe predictors.

Rather than treating all predictors separately, they can be divided into four groups:

1. *schedule and patient descriptors*, including scheduling strategy, specialty, day, operating room, planned order, patient index, expected clinical duration, and priority;

2. *current-time and residual-capacity information*, including elapsed session time and remaining regular capacity;

3. *remaining elective workload*, both for the current operating room and at day level, together with normalized load and position indicators;

4. *observed non-elective information*, constructed only from arrivals that have occurred by the current decision time.

Let t^s_{jk} denote the replay clock immediately before the decision associated with patient i_{jk}. The remaining regular capacity is

$$RC^s_{jk} = \max\{0, H - t^s_{jk}\}.$$

The number of elective surgeries still planned after the current patient in the same operating room is

$$N^{OR}_{jk} = m_j - k,$$

and their remaining expected clinical workload is

$$W^{OR}_{jk} = \sum_{\ell=k+1}^{m_j} \bar{d}_{i_{j\ell}}.$$

Day-level variables provide a broader measure of the elective workload still associated with the current day. In the implementation, the current operating room contributes only patients scheduled after the current position, while the other operating rooms assigned to the same day contribute their complete planned elective lists.

Several normalized indicators are then constructed. Let

$$NE_t^{\exp}$$

denote the total expected workload associated with non-elective arrivals already observed by time t. The operating room load ratio is

$$LR_{jk}^s = \frac{W_{jk}^{OR} + NE_{t_{jk}^s}^{\exp}}{\max\{RC_{jk}^s, 1\}}.$$

The current-case risk ratio is

$$CR_{jk}^s = \frac{\bar{d}_{i_{jk}}}{\max\{RC_{jk}^s, 1\}},$$

while the relative position and elapsed-time ratios are

$$PR_{jk} = \frac{k-1}{\max\{m_j - 1, 1\}},$$

and

$$TR_{jk}^s = \frac{t_{jk}^s}{H}.$$

These variables provide scale-independent measures of how demanding the remaining elective schedule is relative to the available regular capacity.

Observed non-elective information. Four dynamic non-elective quantities are included in the online-safe state:

- the number of non-elective arrivals observed up to the current decision time;
- the sum of their *expected* workloads;
- a proxy for the number of observed non-elective patients whose effective due time has not yet passed;
- the minimum remaining time to an observed non-elective due time.

All four quantities are computed exclusively from events satisfying

$$a_r \leq t.$$

Specifically, the classifier does not receive realized non-elective workload. However, realized quantities and full-day totals are retained in diagnostic datasets for ex-post analysis, but these values are excluded from the online-safe predictor set.

The state representation also includes the non-elective arrival intensity λ, the common elective reference used for its computation, and the number of elective patients in

the weekly schedule. These quantities are known prior to real-time execution and hence do not reveal future scenario realizations.

Time convention and leakage checks. The two numerical time variables stored in the final dataset follow the same convention as the discrete-event simulator: time is measured from the start of the operating session. As a result,

$$08{:}00 \longleftrightarrow 0$$

rather than 480 minutes from midnight. This common convention avoids a train-to-deployment inconsistency between the reconstructed dataset and the simulation model.

A fixed set of online-safe predictors is used throughout the classification experiments. Oracle outcomes, complete future non-elective demand, realized future workload, total oracle overtime, and other ex-post scenario diagnostics are physically excluded from the ML-safe dataset. An explicit leakage check verifies that no forbidden oracle or future-information variable belongs to the predictor set.

5.2 Classification and Policy Construction

Each reachable chronological state is paired with the execute/cancel decision generated by the perfect-information oracle. The learning problem is to approximate this state-to-action mapping using only the 27 online-safe predictors available at the decision point.

The learning problem is formulated as binary classification. Each observation corresponds to an elective decision point reached during the chronological replay described in Section 5.1. The target variable is

$$L_i^s \in \{0, 1\},$$

where $L_i^s = 1$ denotes execution and $L_i^s = 0$ denotes cancellation. The associated feature vector contains only information that would be available at the corresponding decision time. In particular, future non-elective arrivals, future realized surgical durations, and variables derived from the perfect-information solution are excluded from the predictor set.

A fitted classifier produces an execution score

$$p_\theta(\mathbf{x}) = \Pr_\theta(L = 1 \mid \mathbf{x}),$$

where θ denotes the fitted model parameters. Given an execution threshold τ, the resulting decision rule is

$$\widehat{L}(\mathbf{x}; \tau) = \begin{cases} 1 & p_\theta(\mathbf{x}) \geq \tau, \\ 0 & p_\theta(\mathbf{x}) < \tau. \end{cases}$$

The threshold is considered part of the online policy rather than only a statistical post-processing parameter. Lower values of τ make the policy more inclined to treat

elective patients, whereas larger values produce more conservative behavior.

Four classifier families are considered: Logistic Regression, Decision Trees, Random Forests, and XGBoost. The comparison therefore includes both relatively interpretable models and more flexible ensemble approaches. We aim to determine whether additional flexibility of a more complex model leads to better approximation of the oracle decisions and, ultimately, to better operational behavior.

Preprocessing and class imbalance. All classifiers use the 27 online-safe predictors defined during the chronological reconstruction. The feature set is fixed before model fitting and is subject to explicit leakage checks. Variables that include oracle information, realized future workload, or observation identifiers are not used as predictors.

Numerical variables are imputed with the median of the corresponding training feature and then scaled. Categorical variables are imputed with their most frequent training value and encoded using one-hot encoding. Unknown categorical levels encountered outside the fitting sample are addressed without referencing validation data.

The classification problem is strongly imbalanced, since cancellation represents only a small fraction of the reachable decision states. For this reason, class balancing is incorporated during fitting. Logistic Regression and Decision Trees use balanced class weights, Random Forests use balanced weights within the bootstrap samples, and XGBoost is fitted using observation-level weights derived from the class frequencies in the training data.

All preprocessing transformations are estimated from the fitting data and stored together with the corresponding classifier in a common prediction pipeline. The purpose of this stage is to assess how well the alternative classification models generalize to different case-mix instances before their operational consequences are considered. Predictive performance is therefore used to characterize the available model families, but it does not by itself determine the final online policy.

Predictive metrics. Because execution observations constitute the majority class, overall accuracy alone can provide an incomplete description of model quality. For this reason, we also consider in the comparison the cancellation precision, cancellation recall, cancellation F1-score, balanced accuracy, ROC-AUC, and average precision for the cancellation class. The trade-off between identifying oracle cancellations and avoiding unnecessary predicted cancellations is summarized by cancellation F1-score. Balanced accuracy assigns equal importance to both classes. While cancellation average precision focuses directly on discrimination of the minority class, the ranking ability of the execution score independently of a particular threshold is evaluated by ROC-AUC.

The threshold-generation stage does not treat predictive performance as the only selection criterion. Instead, each candidate threshold is passed to the operational validation procedure, so that model complexity and threshold choice are evaluated jointly.

5.3 Operational Policy Selection

Predictive agreement with the oracle does not necessarily imply good performance when a classifier is used as a sequential decision rule. Once deployed, each execute/cancel decision changes the operating-room state encountered at subsequent decision times. For this reason, evaluating a classifier only on a precomputed collection of states would ignore the state-distribution shift induced by its own decisions.

Candidate policies are consequently evaluated on a separate operational-validation subset through an event-driven operational validation procedure. The classifier is queried during schedule execution, and the system state is updated after each elective or non-elective service event. Different policies may thus reach different chronological states even when evaluated under the same underlying stochastic realization.

Event-driven policy evaluation. Within each trajectory, non-elective arrivals are revealed only when their arrival time is reached. Whenever an operating room becomes available, waiting non-elective patients take precedence over elective activity and are processed according to the urgency-based queue rule described in Section 5.1. If no non-elective patient is waiting, the state associated with the next elective patient is constructed from the current trajectory and passed to the classifier.

If

$$p_\theta(\mathbf{x}) \geq \tau,$$

the elective procedure is executed and the operating-room clock advances according to its realized duration and turnover time. Otherwise, the procedure is canceled. After the first elective cancellation in a room, the remaining elective suffix is canceled according to the same prefix structure used when constructing the oracle labels.

Future non-elective arrivals are never made available to the classifier. As a result, the validation procedure therefore preserves the information structure required for online deployment while allowing previous learned decisions to affect all subsequent states.

Joint policy-selection criterion. A single model–threshold policy is required for the final evaluation, although the underlying elective schedules may be generated according to either the patient-centered or the facility-centered strategy. Candidate selection therefore combines information from both scheduling settings.

The operational component preserves the hierarchical objectives used in the optimization framework. Under the patient-centered strategy, the primary measure is the mean executed patient priority, while the secondary criterion combines cancellations and regular overtime. Under the facility-centered strategy, the primary measure is the realized duration of executed elective activity, while the secondary criterion combines regular overtime and cancellations.

Rather than combining these quantities directly despite their different scales, candidates are compared through normalized ranks. Let r_c^{op} denote the normalized operational rank of candidate c, obtained while preserving the patient-centered and facility-centered

objective hierarchy. Let $r_c^{F_1}$ and r_c^{acc} denote the corresponding normalized ranks for cancellation F1-score and accuracy on the operational validation data.

The final policy-selection score is

$$S_c = \alpha\, r_c^{\text{op}} + \beta\, r_c^{F_1} + \gamma\, r_c^{\text{acc}},$$

where $\alpha, \beta, \gamma \geq 0$ and $\alpha + \beta + \gamma = 1$. Lower values are preferred. The weights used in the computational experiments are reported in Chapter 7. Predictive comparison and operational validation are kept as separate stages, and predictive-validation results are not reused as threshold-specific final-selection metrics.

We evaluate the threshold-specific F1-score and accuracy entering the selection criterion at the same threshold used in the dynamic rollout. The experimental weighting scheme and the instance-level partition used for policy selection are specified in Chapter 7.

5.4 Final Refit

After the model configuration and threshold have been selected, no additional tuning is performed. The selected model configuration is refitted on the development data, while the execution threshold remains fixed.

This refit allows the final classifier to use all information available from the development instances without altering the policy-selection decision.

Chapter 6

Discrete-Event Simulation

After an online policy has been put into action, each choice made has an effect on the state that will be reached at later decision points. When a patient is operated on, the operating-room clock is advanced by the actual surgical duration together with the turnover time, while if a patient is canceled, then the remaining sequence of elective patients is altered. At the same time, non-elective patients arrive dynamically, and this may cause a delay in later elective activities. As a result, a policy may reach states which were not met along the oracle trajectory.

For this reason, the final evaluation is performed through a discrete-event simulation (DES) in which the policy under evaluation is queried sequentially during schedule execution. The simulation provides a closed-loop evaluation of the complete decision process and allows the learned policy to be compared with alternative online decision rules under common stochastic realizations.

6.1 Simulation Logic

The simulation state records the current time of each operating room, the remaining elective sequence, the non-elective arrivals already observed, and the corresponding waiting queue. We measured time in minutes from the beginning of each daily operating-room session, with $t = 0$ corresponding to the start of regular activity.

An illustration of the mechanism on an operating-room sequence is shown in Figure 1. The release time of the room is changed by a delay in an elective procedure. The resulting clock shift changes the state at which the next elective decision is made. If the policy subsequently cancels an elective case, the current case and the remaining elective suffix in that operating room are removed by the prefix-feasibility rule.

Chronological event processing. At each operating-room release time, all non-elective arrivals observed up to the current simulation time are added to the waiting queue. Non-elective patients have priority over elective patients and are ordered according to earliest due time. Hence, if at least one non-elective patient is waiting when

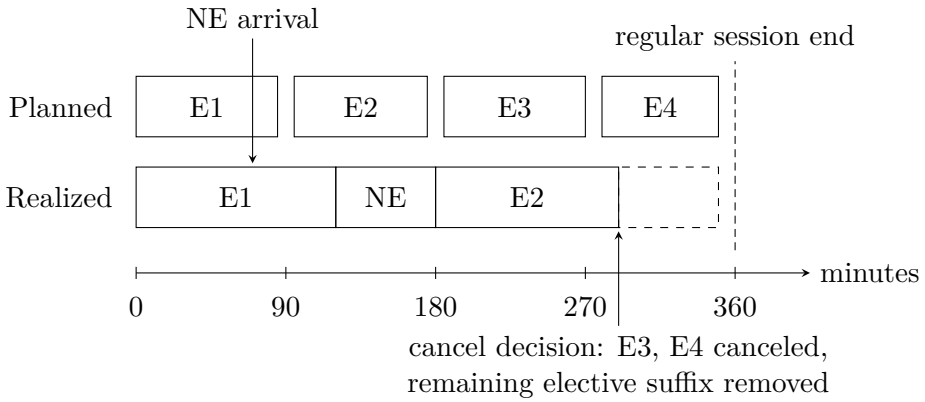

Figure 1: Effect of stochastic execution on a planned operating-room sequence. When a non-elective patient arrives while the room is busy, it waits until the room is released, then is served before the next elective patient; the remaining elective suffix is removed by a later elective cancellation.

a room becomes available, the most urgent waiting patient is assigned before any new elective procedure is considered.

If no non-elective patient is waiting, the next elective patient in the planned sequence becomes eligible for an online decision. The current state is reconstructed using exactly the same online-safe feature definition adopted during model training. In particular, the classifier does not receive information about future non-elective arrivals or future realized surgical durations.

For an elective patient i considered at time t, the learned policy computes

$$p_\theta(\mathbf{x}_{i,t}) = \Pr_\theta\left(L_{i,t} = 1 \mid \mathbf{x}_{i,t}\right).$$

The patient is executed if

$$p_\theta(\mathbf{x}_{i,t}) \geq \tau,$$

and canceled otherwise.

When the patient is executed, the operating-room release time advances according to the realized clinical duration and the applicable turnover time. If the patient is canceled, the prefix-feasibility rule is activated: the current elective case and all subsequent elective cases planned in the same operating room are canceled. The operating room may still serve non-elective demand arriving later in the day.

This mechanism is important because the classifier is queried only on states that are actually reached under its previous decisions. Thus, the DES differs fundamentally from the static classification evaluation, where states are fixed before predictions are produced.

6.2 Decision Policies and Performance Measures

Elective decision policies. The simulator evaluates execute/cancel policies using the same state-transition logic. A learned policy executes an elective patient whenever its execution score satisfies

$$p_\theta(\mathbf{x}) \geq \tau,$$

and cancels it otherwise. Hence, the same fitted classifier can be evaluated at different execution thresholds without changing the simulation mechanism.

An *always-execute* rule provides an aggressive benchmark in which execution occurs whenever an elective patient reaches a decision point.

A second benchmark is the *expected-capacity rule*: at decision time t, the next elective patient is executed only if its expected completion time satisfies

$$t + \bar{d}_i + \tau_i \leq B + O^{\max},$$

where $\bar{d}_i$ is its expected clinical duration and τ_i is the expected turnover time, where applicable. This rule uses the current system time and the expected requirements of the current elective patient, but does not anticipate future non-elective arrivals.

The exact classifier thresholds and policy set used in the computational experiments are reported in Chapter 7.

Performance measures. Policy performance is evaluated using both elective and non-elective outcomes.

For the elective component, the main measures are the number of canceled patients, the cumulative priority of executed patients, and the realized duration of executed elective activity. The relevance of the last two measures depends on the scheduling strategy: executed priority is the main patient-centered indicator, whereas executed realized duration is the main facility-centered indicator.

Let B denote the regular session length and $O^{\max}$ the regular overtime allowance. For each operating-room session with final release time C_{jd} then overtime is decomposed as

$$OT_{jd} = \max\{0, C_{jd} - B\},$$
$$OT_{jd}^{\text{reg}} = \min\{OT_{jd}, O^{\max}\},$$

and

$$OT_{jd}^{\text{extra}} = \max\{0, OT_{jd} - O^{\max}\}.$$

Adding up these values for all the operating-room sessions, we obtained weekly regular and extra overtime. Non-elective case performance is described by means of the mean and the 95th-percentile waiting time, the number of service-limit violations, and the proportion of patients who were seen within their urgency-specific time limits.

Given that common random numbers are used, differences between policies are also computed for each replication. Let $Y_{c,r}$ denote the value of a performance measure for comparator c in replication r, and let $Y_{\star,r}$ denote the corresponding result for the selected

policy. Paired differences provide estimates of

$$\Delta_c = \mathbb{E}\left[Y_\star - Y_c\right],$$

together with 95% confidence intervals.

Chapter 7

Instance Generation and Data Analysis

The algorithms must be tested in a highly diverse and realistic stochastic environment to properly validate the optimization-guided learning policy. We next describe the process for generating individual cases, which is based on the procedural time distributions obtained by Korzhenevich and Zander [KZ24] from the *German Operating Room Benchmarking Initiative*. The aim is to test the proposed sequential decision pipeline through genuine grounded peri-operative variability, asymmetric duration distributions, and competing non-elective demand.

The chapter first introduces the operating room setting and the available data, and then describes the procedure used to generate the synthetic case-mix instances. The final sections examine the variability of the sampled clinical durations and discuss the main sources of uncertainty and the limitations of the data.

7.1 Empirical Data and Operating-Room Setting

The data underlying this collection refer to the year 2019, selected as the latest available period unaffected by COVID-19-related disruptions, comprising 2,035,126 recorded surgeries from 212 German, Austrian, and Swiss hospitals. Following the German Perioperative Procedural Time Glossary (GPPTG), the dataset defines standardized perioperative process time points covering patient logistics, operating room logistics, anesthesia, and surgical activity. After data-quality and plausibility checks, the source derives a ready-to-use collection of surgical case mixes and fitted process-time distributions for planning applications.

The computational experiments developed in this thesis focus on hospitals classified as *Basic and Regular Care* and on three surgical specialties: General Surgery, Otolaryngology, and Gynecology & Obstetrics. The operating room system consists of elective surgical procedures assigned to rooms and performed sequentially during a regular operating session. Each surgical case requires a sequence of peri-operative activities, while a turnover period is needed between consecutive cases in the same operating room.

During execution, actual surgical durations may differ from their expected values. As these deviations accumulate along an operating room sequence, the planned completion time may shift, and the remaining elective workload may no longer fit comfortably within the available capacity. Moreover, in the extended setting considered later in the thesis, stochastic non-elective patients compete with elective cases for the same operating room resources. These features create the evolving environment in which the execute/cancel policy developed in Chapter 5 is subsequently evaluated. Rather than using the original hospital-level records directly, the computational analysis presented in the thesis uses the processed data collection derived from the benchmarking initiative. In particular, the implementation relies on the case-mix frequencies and the fitted probability distributions supplied with the data collection. For each surgical record, information is available on the hospital, level of care, surgical specialty, surgery date, operating room, and anonymized patient stay. Additional variables, when recorded, include the main surgical procedure identified by its OPS code, patient type, anesthesia type, urgency category, and further peri-operative timestamps.

The preprocessed dataset retains different hospital levels of care—Basic and Regular Care, Specialized Care, Maximum Care, and University Clinics—surgical specialties—General Surgery, Trauma Surgery, Otolaryngology, and Gynecology & Obstetrics—and types of patient—inpatient and outpatient. For each combination of hospital level, specialty, and patient type, the case-mix also reports the observed frequency of individual OPS codes. The corresponding process-time files provide fitted distributions for the surgical phases associated with each procedure.

Although the complete data collection contains several hospital and specialty settings, the computational experiments in this thesis are restricted to the *Basic and Regular Care* hospital level and to General Surgery, Otolaryngology, and Gynecology and Obstetrics. This restriction keeps the experimental setting consistent across the optimization, learning, and simulation stages while retaining substantial heterogeneity in procedure types and durations.

Process time variables. The surgical process is represented through five consecutive clinical phases: anesthesia induction, surgical lead-in, incision-to-closure time, surgical lead-out, and anesthesia emergence. For each combination of hospital level, surgical specialty, patient type, and OPS code, the dataset provides a fitted probability distribution for each phase. The candidate distribution families are lognormal, gamma, and Weibull.

Operating room cleaning duration is modeled from these clinical phases. Its distribution depends on hospital level, surgical specialty, and patient type but not on the OPS code. In the computational model, cleaning time is interpreted as the turnover component between consecutive surgeries rather than as part of the clinical duration of a procedure.

This distinction is maintained throughout the optimization framework: surgical duration represents the time associated with the clinical phases of a patient, while turnover is handled separately when consecutive cases are scheduled in the same operating room.

7.2 Synthetic Case-Mix Generation

The processed benchmarking data provide distributions and case-mix frequencies rather than the controlled planning instances required for the computational experiments.

Synthetic case-mix instances are therefore generated by sampling from these empirical inputs. This makes it possible to construct multiple comparable scheduling instances while preserving the procedure mix and duration characteristics observed in the source data.

Each synthetic instance contains 200 surgical elective patients belonging to a single surgical specialty. The composition is fixed at 57 inpatients and 143 outpatients, corresponding to an outpatient share of 71.5%. This choice is consistent with the high outpatient proportions reported by Omling et al. [Oml+18], while keeping the inpatient–outpatient composition constant across instances so that differences between experiments are not driven by changes in patient-type composition.

For a fixed hospital level ℓ, specialty s, and patient type t, an OPS code is sampled according to its empirical frequency in the case-mix data. If $N_{\ell sto}$ denotes the number of observations associated with OPS code o, the sampling probability is

$$p_{\ell st}(o) = \frac{N_{\ell sto}}{\sum_{o' \in \mathcal{O}_{\ell st}} N_{\ell sto'}}.$$

Once the OPS code has been sampled, the corresponding fitted distributions are retrieved for the five clinical process phases. Let

$$\mathcal{P} = \{\text{induction, lead-in, incision-to-closure, lead-out, emergence}\}.$$

For the patient i, the clinical duration is represented as

$$D_i = \sum_{r \in \mathcal{P}} D_{ir},$$

where D_{ir} denotes the duration of phase r. In the synthetic generation procedure, the phase durations are sampled independently from their fitted procedure-specific distributions. Consequently, the expected clinical duration stored for the patient i is

$$\bar{D}_i = \sum_{r \in \mathcal{P}} \mathbb{E}[D_{ir}].$$

The cleaning time is generated separately and provides the turnover component associated with the patient. The generated case-mix files retain both expected and sampled clinical and turnover durations.

Each elective patient is assigned a synthetic normalized priority score, as the source data lack a patient-level priority measure appropriate for the patient-centered scheduling objective. We must interpret it as an experimental relative priority with the goal of introducing heterogeneity in patient importance and to allow the patient-centered scheduling strategy to distinguish among otherwise schedulable elective cases.

Three surgical specialties are considered: General Surgery, Otolaryngology, and Gynecology & Obstetrics. Twelve case-mix instances are used for each specialty, giving a total of 36 synthetic case-mix instances and $36 \times 200 = 7,200$ elective patients. Instances 1–10 constitute the development set used during model construction and policy selection. Instances 11–12 are not used for model, hyperparameter, or threshold selection and are retained for the corrected final re-evaluation described in Chapters 8.

7.3 Exploratory Data Analysis

The descriptive analysis considers the 7,200 elective patients contained in the 36 synthetic case-mix instances generated for the three surgical specialties. For descriptive purposes, all twelve instances per specialty are included in this section, including the two instances later reserved for final testing. These summary statistics are used only to characterize the generated experimental data and do not enter model fitting, hyperparameter selection, or policy selection.

Since clinical duration and operating room turnover are modeled separately in the subsequent optimization framework, the analysis below focuses on the sampled clinical duration of each patient, defined as the sum of the five peri-operative clinical phases introduced in Section 7.1. Accordingly, this section excludes cleaning time from the duration statistics and treats it separately as turnover time in the optimization models.

Descriptive Statistics. Table 7.1 summarizes the sampled clinical durations across the three specialties.

Table 7.1: Summary statistics of sampled clinical durations by specialty.

Specialty	Mean (min)	Std (min)	Q1 (min)	Median (min)	Q3 (min)	Max (min)
General Surgery	72.31	42.18	45.81	62.02	85.37	537.97
Gynecology & Obstetrics	56.17	36.58	32.23	43.98	68.51	392.73
Otolaryngology	38.92	21.51	25.52	34.24	46.05	224.18

Clear differences can be observed both in typical procedure duration and in absolute dispersion. General Surgery has the longest procedures on average, with a mean clinical duration of 72.31 minutes, followed by Gynecology and Obstetrics with 56.17 minutes. Otolaryngology procedures are considerably shorter, with an average duration of 38.92 minutes. The same ordering is observed for the medians. General Surgery also shows the largest absolute standard deviation, equal to 42.18 minutes, and contains the longest sampled procedure, with a duration of 537.97 minutes. However, absolute dispersion should be interpreted together with the typical duration of each specialty. As shown below, the coefficient of variation provides a complementary measure of variability relative to the corresponding mean.

For all three specialties, the mean is higher than the median and the maximum duration lies far above the third quartile. These features indicate positively skewed duration distributions, with a relatively small number of considerably longer procedures.

Differences by Patient Type. Duration heterogeneity is also observed between inpatient and outpatient procedures. Table 7.2 reports the corresponding descriptive statistics within each specialty.

Table 7.2: Sampled clinical duration by surgical specialty and patient type.

Specialty	Patient type	Count	Mean (min)	Std (min)	Median (min)
General Surgery	Inpatient	684	106.86	58.02	94.26
	Outpatient	1716	58.54	21.96	54.81
Gynecology & Obstetrics	Inpatient	684	92.36	45.28	82.48
	Outpatient	1716	41.74	18.03	37.43
Otolaryngology	Inpatient	684	51.74	30.78	43.44
	Outpatient	1716	33.80	13.34	31.60

Inpatient procedures are longer on average in all three specialties. The difference is particularly pronounced in General Surgery and Gynecology and Obstetrics, where the mean clinical duration of inpatient procedures exceeds that of outpatient procedures by approximately 48 and 51 minutes, respectively. A smaller but still clear difference is observed for Otolaryngology. Inpatient procedures also exhibit greater absolute dispersion within each specialty.

These differences show that the variability of the generated case mixes is not determined by surgical specialty alone. Patient type also contributes to the duration profile of the instances and is consequently retained as a patient-level characteristic throughout the computational framework.

Distribution of Surgery Durations. Figure 2 shows the distributions of sampled clinical durations for the three specialties. The histograms highlight the right-skewed structure already suggested by the descriptive statistics: most procedures are concentrated in the lower part of the duration range, while progressively fewer observations extend into a long right tail.

The cross-specialty differences are shown more directly in Figure 3. General Surgery exhibits the widest absolute range and the highest typical duration, whereas Otolaryngology procedures are generally shorter. Gynecology & Obstetrics occupies an intermediate position in terms of absolute duration but, as shown below, presents the largest variability relative to its mean.

Heterogeneity Across Specialties and Patient Types. The processed benchmarking data distinguish several hospital levels of care, specialties, and patient types. The computational study developed in this thesis deliberately fixes the hospital setting to Basic and Regular Care, while varying the surgical specialty and the composition of the synthetic case mixes.

The resulting duration profiles nevertheless remain markedly heterogeneous across both specialties and patient types. General Surgery, Otolaryngology, and Gynecology

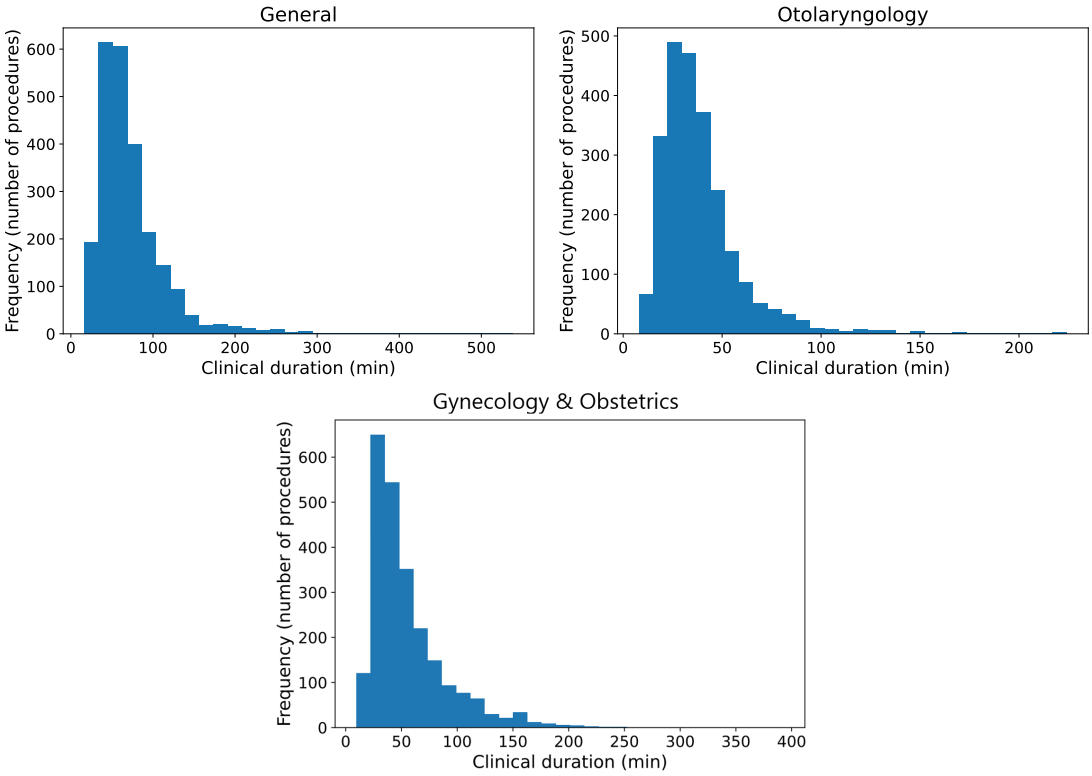

Figure 2: Distribution of sampled clinical durations in General Surgery, Otolaryngology and Gynecology & Obstetrics across the 36 synthetic case-mix instances. Clinical duration excludes operating room turnover time.

& Obstetrics differ substantially in typical clinical duration and dispersion, as shown in Table 7.1 and Figures 2–3. Within each specialty, inpatient procedures are also consistently longer and more dispersed than outpatient procedures, as shown in Table 7.2. The subsequent optimization and learning experiments therefore operate on case mixes containing substantial heterogeneity even though the hospital level is kept fixed.

Variability of Surgery Durations. Uncertainty in surgical duration is one of the main stochastic components of the operating room system considered in this thesis. The fitted distributions provided by Korzhenevich and Zander [KZ24] include lognormal, gamma, and Weibull families, reflecting the asymmetric and right-tailed duration patterns observed in the underlying surgical process data.

In the computational experiments, uncertainty is represented at the level of the individual clinical phases. For a patient i, the realized clinical duration is given by

$$D_i = \sum_{r \in \mathcal{P}} D_{ir},$$

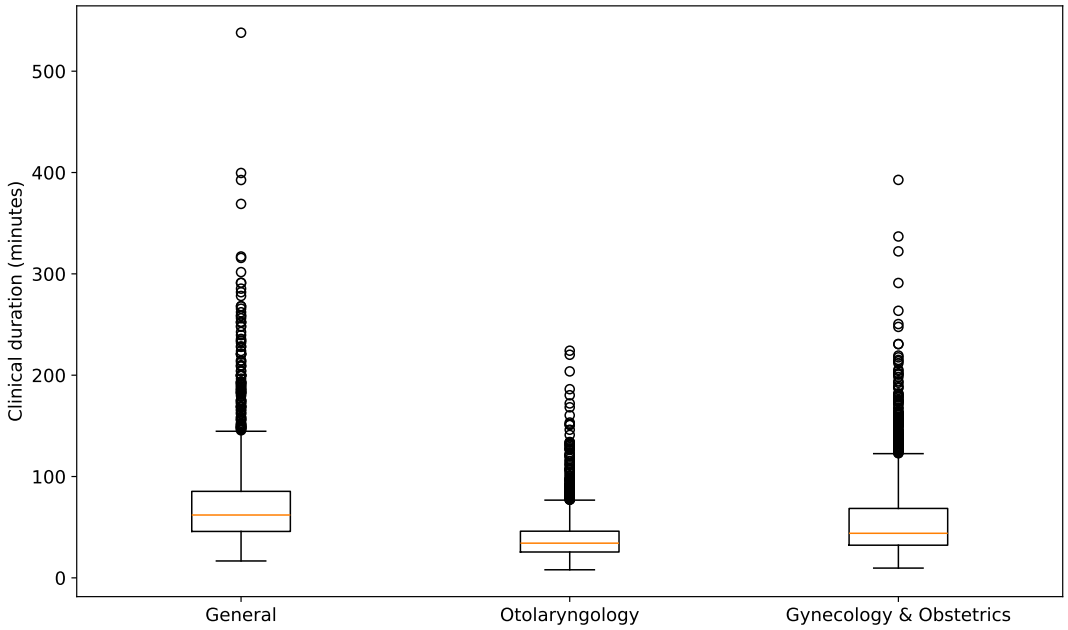

Figure 3: Comparison of sampled clinical duration distributions across specialties.

where $\mathcal{P}$ denotes the five clinical phases defined in Section 7.2. Monte Carlo sampling from these patient-specific distributions subsequently produces different realizations of the same planned surgical workload.

As a descriptive measure of relative dispersion in the generated case mixes, the coefficient of variation is computed separately for each instance. If $\bar{D}_j$ and s_j denote respectively the sample mean and sample standard deviation of the clinical durations in case-mix instance j, its coefficient of variation is

$$CV_j = \frac{s_j}{\bar{D}_j}.$$

Unlike the standard deviation, the coefficient of variation expresses dispersion relative to the average duration and thereby facilitates comparisons across specialties characterized by different duration scales.

Figure 4 reports the distribution of the case-mix-level coefficients of variation, while Table 7.3 summarizes the corresponding values.

Gynecology & Obstetrics has the highest average relative variability, with a mean coefficient of variation of approximately 0.65. General Surgery follows with an average CV of approximately 0.58, while Otolaryngology has the lowest average value, approximately 0.55. Nevertheless, substantial relative dispersion is present in every specialty: across all 36 generated instances, the coefficient of variation ranges from approximately 0.46 to 0.71.

These results support the use of a scenario-based representation of surgical durations

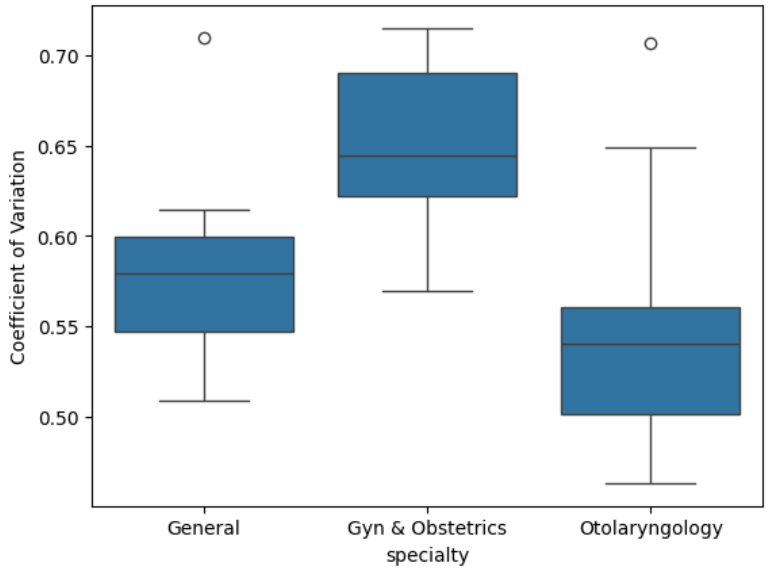

Figure 4: Distribution of the case-mix-level coefficient of variation of sampled clinical durations across specialties. Each specialty contains twelve independently generated case-mix instances.

Table 7.3: Summary statistics of the case-mix-level coefficient of variation of sampled clinical durations.

Specialty	Mean CV	Std CV	Min CV	Max CV
General Surgery	0.579	0.054	0.509	0.710
Otolaryngology	0.548	0.073	0.463	0.707
Gynecology & Obstetrics	0.650	0.047	0.569	0.715

in the stochastic scheduling models developed in Chapter 4. A schedule constructed only from expected durations would not represent the range of possible duration realizations that can occur during execution.

Non-Elective Demand. Surgical-duration uncertainty is not the only stochastic component considered in the final framework. Operating room capacity may also be required by non-elective patients whose arrival times and workload are not known when the elective schedule is constructed.

The German benchmarking data used to generate the elective case mixes do not provide the complete arrival-process information required to estimate the non-elective demand model adopted in this thesis. Consequently, non-elective demand is introduced as an exogenous stochastic component in Chapter 4, where daily arrivals are generated through a Poisson process and compete with elective surgeries for shared operating room

capacity.

This distinction is important when interpreting the experiments. The distributions of elective surgical durations are directly grounded in the processed German benchmarking data, whereas the non-elective arrival process represents an additional modeling assumption designed to introduce stochastic urgent demand into the operating room system.

Schedule Delays and Propagation Effects. Uncertainty in individual durations becomes operationally relevant because surgeries assigned to the same operating room are performed sequentially. A procedure that lasts longer than expected delays the beginning of the next scheduled case, and these deviations may accumulate over the course of the session.

Consequently, uncertainty cannot be interpreted only at the level of individual patients. What matters for the online decision problem is also the state reached after the previous realized durations and non-elective workload have consumed part of the available capacity. As the operating session progresses, the residual capacity may therefore differ substantially from what was expected when the schedule was constructed.

This propagation mechanism is central to the execute/cancel decision considered in the subsequent chapters: performing an additional elective surgery preserves elective activity but may increase overtime, whereas canceling it protects residual capacity at the cost of postponing the patient.

Data Limitations and Unobserved Uncertainty. The empirical and synthetic nature of the experimental framework introduces several limitations that should be considered when interpreting the computational results.

First, the processed benchmarking data aggregate observations from several hospitals, surgical teams, and organizational environments. Institution-specific factors that may influence surgical performance are therefore not modeled explicitly. Furthermore, the present study focuses only on the Basic and Regular Care hospital level, so the numerical results should not be interpreted as directly representative of all hospital settings contained in the original benchmarking initiative.

Second, information on waiting lists, detailed staffing levels, future non-elective arrivals, and complete hospital-specific planning rules is not available from the data source used in this thesis. The fixed inpatient–outpatient composition of the synthetic instances is taken from external literature rather than estimated specifically for the Basic and Regular Care setting.

Third, patient priorities are synthetically generated rather than derived from a validated clinical urgency scale. They should therefore be interpreted only as relative experimental weights used to distinguish the patient-centered scheduling objective.

Finally, the phase-specific fitted distributions are sampled independently in the synthetic generation procedure. Possible dependence between different peri-operative phases of the same surgery is therefore not explicitly represented. Similarly, the stochastic non-elective arrival process introduced later in the framework is a modeling component rather

than an empirically estimated arrival model for the hospitals contained in the German dataset.

These limitations do not prevent the framework from being used to compare alternative scheduling and online decision policies under controlled stochastic conditions, but they restrict the extent to which the numerical results can be directly generalized to a specific hospital without further calibration.

7.4 Experimental Setup

The study considers the three surgical specialties introduced above: General Surgery, Otolaryngology, and Gynecology & Obstetrics. Twelve independently generated case-mix instances are available for each specialty, and each instance contains 200 elective patients.

Two scheduling strategies are evaluated for every instance–specialty pair. The *patient-centered* strategy prioritizes the inclusion of patients with higher synthetic priority scores, whereas the *facility-centered* strategy favors the amount of elective clinical workload scheduled within the available operating room capacity. Hence, the complete study contains $12 \times 3 \times 2 = 72$ strategy-specific scheduling configurations.

The partition into development and evaluation subsets is performed at the case-mix level rather than at the level of individual observations. Consequently, all scenarios and all decision states generated from the same case-mix instance remain in the same subset. This prevents observations derived from the same underlying synthetic patient population from appearing in both model fitting and evaluation. The role of the instances is summarized in Table 7.4.

Table 7.4: Instance-level partition used in the optimization-guided learning pipeline.

Component	Model fitting	Predictive validation	Operational validation	Final evaluation
Case-mix instances	1–6	7–8	9–10	11–12
Specialties	3	3	3	3
Scheduling strategies	2	2	2	2
Strategy-specific schedules	36	12	12	12
Purpose	Estimator fitting	Predictive comparison	Model–threshold selection	Final re-evaluation

Instances 1–6 are used to fit the candidate classification models. Instances 7–8 provide an independent predictive-validation subset for comparing model configurations, while instances 9–10 are reserved for the operational selection of the model and execution threshold. After this selection has been completed, the chosen configuration is refitted on instances 1–10. Instances 11–12 remain excluded from model, hyperparameter, and threshold selection and are used for the final re-evaluation reported in Chapter 8.

The same instance-level separation is maintained throughout scenario generation, oracle label generation, and chronological state reconstruction.

SAA and solver settings. The initial scheduling stage uses $N = 70$ Monte Carlo duration scenarios for each instance-specialty pair. This same scenario pool is used for the two scheduling strategies so that their comparison is not affected by different random duration samples. Rather than solving a single stochastic program on all 70 scenarios, the implementation uses five SAA replications. The scenario indices are randomly shuffled and divided into $K = 5$ non-overlapping folds, each containing 14 scenarios. A stochastic scheduling problem is solved independently on each fold, producing five candidate schedules.

Computational settings. Each fold is assigned a maximum solution time of 720 seconds and a relative MIP gap tolerance of 5%. 80% of the total time budget is initially allocated to the deterministic primary-objective phase: $0.8 \times 720 = 576$ seconds.

For the oracle optimization, we set the penalty, tolerance, and tie-breaking parameters to

$$\zeta_{\mathrm{NE}} = 10^4, \qquad \varepsilon_F = 10^{-6}, \qquad \eta_P = \eta_U = 10^{-4}.$$

We defined the strategy-specific tolerance as

$$\varepsilon_P = 10^{-6} + 10^{-3}|P^{s,*}|, \qquad \varepsilon_U = 10^{-6} + 10^{-3}|U^{s,*}|.$$

If this phase terminates before reaching the time limit, the remaining time is reallocated to the scenario-dependent recourse phase: this means that the maximum computational budget for the five candidate problems under one strategy-specific configuration is $5 \times 720 = 3600$ seconds.

For each strategy-specific configuration, we solve the weighted reduced-SAA formulation once over the $K = 10$ representative scenarios, with a time limit of 720 seconds and a relative MIP gap of 5%. We cluster the 1000 scenarios once before solving the 60 configuration-specific reduced-SAA problems. We then recorded the computational times during the experiments.

The most suitable primary value that can be obtained in the second phase is the value of the incumbent solution found within the given time period, not necessarily the global optimum, since the primary phase might end with a nonzero MIP gap; it is for this reason that the second phase has been implemented in such a way as to retain at least the primary incumbent value and to optimize the strategy-specific recourse objective.

The 70 scenarios used in this initial SAA procedure should not be confused with the larger Monte Carlo sample introduced in Section 4.2. The latter contains 1000 realizations per instance–specialty pair and is used for scenario reduction, out-of-sample evaluation, and oracle label generation.

Scenario-reduction and non-elective settings. For each instance-specialty pair, 1000 independent Monte Carlo realizations of elective clinical durations are generated

from the patient-specific distributions described in Chapter 7. The same scenario pool is used for the facility-centered and patient-centered strategies.

No standardization is performed in the final experiments. The number of clusters is fixed to $K = 10$, with 20 initializations of the K-means algorithm.

The target non-elective share used in the experiments is $q = 0.15$. To avoid making the non-elective arrival process depend on the scheduling strategy, n_{ref} is computed as the rounded mean of the numbers of elective patients scheduled by the original facility-centered and patient-centered SAA_N solutions. Consequently, the same value of λ and the same non-elective scenario realizations are used for both strategies within an instance–specialty pair.

Each non-elective patient belongs to one of four urgency classes. The probabilities and response-time limits used in the experiments are reported in Table 7.5.

Table 7.5: Non-elective urgency classes used in the stochastic experiments.

Class	Probability	Time limit (min)
Trauma	0.20	30
Emergent	0.40	90
Urgent	0.30	180
Add-on	0.10	1440

The recourse problems were solved using a target MIP gap of 1% and a finite time limit for computation. Consequently, the requested gap was not necessarily attained for every scenario batch.

Machine-learning settings. A total of 29 model configurations are considered. The alternative configurations are compared on case-mix instances that are completely separated from the fitting instances rather than through a random observation-level split. Table 7.6 reports the hyperparameter grid.

Table 7.6: Candidate classifier configurations.

Model	Hyperparameters	Config.
Logistic Regression	$C \in \{0.01, 0.1, 1, 10\}$	4
Decision Tree	depth $\in \{5, 8, 12\}$, min. leaf $\in \{100, 300, 500\}$	9
Random Forest	trees $\in \{100, 200\}$, depth $\in \{10, 18\}$, min. leaf $\in \{100, 300\}$	8
XGBoost	depth $\in \{3, 5\}$, learning rate $\in \{0.03, 0.05\}$, trees $\in \{200, 400\}$	8
Total		29

The relatively large minimum-leaf values used for the tree-based models limit excessively local decision rules in a dataset containing several million related chronological observations. For XGBoost, both row and column subsampling rates are set to 0.8.

Model development follows the instance-level partition introduced above. Candidate model configurations are fitted on instances 1–6 and compared predictively on instances

7–8. Instances 9–10 are then used to construct and compare alternative model–threshold policies according to their predictive and operational performance. Once the final policy has been selected, the corresponding model configuration is refitted on instances 1–10 while keeping the selected threshold fixed. Instances 11–12 are not used for model, hyperparameter, or threshold selection and are retained for the corrected final re-evaluation.

For all 29 of the fitted configurations, the threshold behavior is examined using instances 9–10. The possible operating points include the standard threshold of 0.50, thresholds linked with the favorable diagnostic classification criteria, and quantile thresholds that are specific to the model and have been selected to cover a range of the raw predicted cancellation rates.

Any duplicate combinations of model and threshold are eliminated, so the final screening set consists of 401 different candidate policies, where a policy is defined by the model family, its hyperparameter configuration, and the execution threshold.

Policy-selection weights. The weights assigned by the experimental joint score are 0.50 for the normalized operational rank, 0.25 for the cancellation F1-score, and 0.25 for accuracy. These weights are the result of an experimental choice made in order to select a policy, not something that has been obtained from a clinical cost model. Half of the total weight is allocated to operational behavior in order to highlight the significance of sequential decision-making as compared to isolated classification. The cancellation F1-score is included in order to deal with the minority cancellation class in the case of class imbalance, whereas accuracy serves to penalize overall disagreement with the oracle.

To assess whether the final policy selection is overly sensitive to this particular weighting specification, we also performed a local sensitivity analysis on instances 9–10. The weights are moderately perturbed around the baseline configuration while preserving their sum equal to one. The following seven specifications are considered:

$$(w_{\text{op}}, w_{F_1}, w_{\text{acc}}) \in \left\{ \begin{array}{lll} (0.45, 0.25, 0.30), & (0.45, 0.30, 0.25), & (0.50, 0.20, 0.30), \\ (0.50, 0.25, 0.25), & (0.50, 0.30, 0.20), & (0.55, 0.20, 0.25), \\ (0.55, 0.25, 0.20) & & \end{array} \right\}.$$

For every specification, the same 401 candidate policies, normalized ranks, and tie-breaking rules are used. No model is refitted and no information from instances 11–12 is used in this sensitivity analysis.

Operational-validation settings. For every candidate policy, 100 stochastic realizations are considered for each combination of the two operational-validation instances, three surgical specialties, and two scheduling strategies. Accordingly, each candidate is evaluated over $2 \times 3 \times 2 \times 100 = 1200$ strategy-specific weekly trajectories. Since 401 model–threshold policies are included in the screening set, the complete operational comparison contains $401 \times 1200 = 481{,}200$ candidate–trajectory evaluations.

Final simulation settings. The final simulation uses the corrected case-mix instances 11–12, which are not used for model, hyperparameter, or threshold selection. Thus, the machine-learning policy is fixed before the simulation is performed.

The final simulation uses the policy selected through the procedure described in Chapter 5. The corresponding classifier is refitted on instances 1–10 while keeping the selected execution threshold fixed. The selected policy is compared with the same refitted classifier at the conventional threshold $\tau = 0.50$, the *always-execute* rule, and the *expected-capacity* rule defined in Chapter 6. The selected model configuration and threshold are reported in Chapter 8.

No model parameter or decision threshold is modified during the final dynamic evaluation.

For each combination of instance and specialty, 1000 independent stochastic replications are generated. The experiment considers

$$2 \text{ instances} \times 3 \text{ specialties} \times 2 \text{ scheduling strategies} \times 4 \text{ online policies} \times 1000 \text{ replications},$$

for a total of 48,000 strategy–policy weekly trajectories.

For each instance–specialty–replication combination, the same realized elective durations and the same non-elective arrival realization are reused across all policies and both scheduling strategies. This common-random-number design ensures that differences between policies are not generated by different stochastic inputs and makes it possible to perform paired comparisons.

Each simulated week contains five working days. A regular operating-room session lasts $B = 360$ minutes and an additional regular overtime allowance of $O^{\max} = 30$ minutes is considered. When activity is beyond 390 minutes, we recorded it separately as extra overtime.

We generate elective surgical durations using the same patient-duration mechanism adopted throughout the computational pipeline; instead, non-elective demand follows the Poisson-based construction introduced in Section 4.2. Arrival times are allocated throughout the regular session once the daily number of arrivals is given, and non-elective patients are categorized as trauma, emergent, urgent, or add-on cases and inherit the corresponding service-time limits [DA19].

The simulation extends beyond the scheduled end of the operating-room session until all generated non-elective patients have been treated. As a result, the absence of unserved non-elective patients reflects a feature of the simulation design rather than an inherent advantage of any specific policy. Policy differences in non-elective service are measured through waiting times, deadline violations, and overtime.

Figure 5 summarizes the workflow.

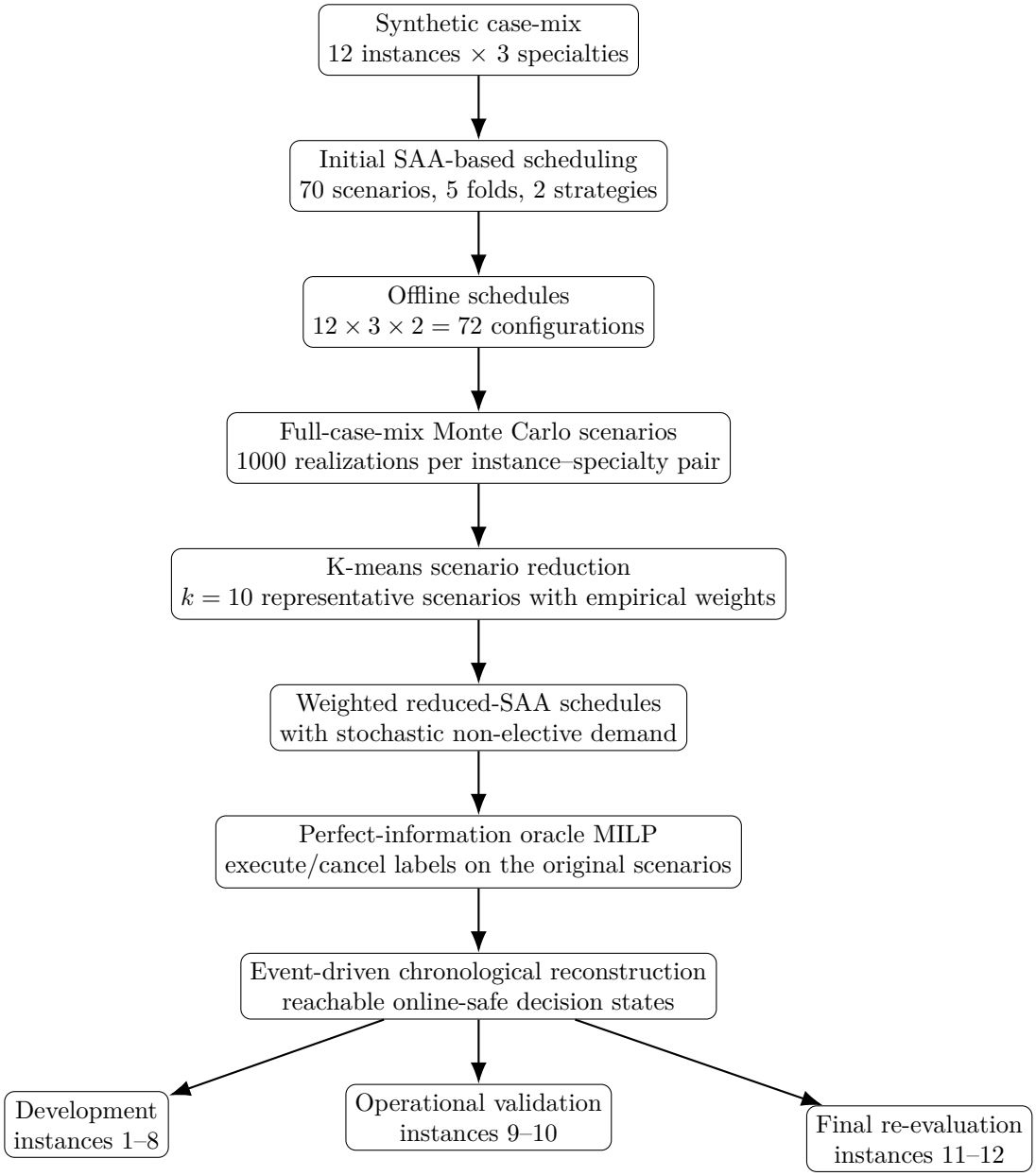

Figure 5: Overview of the offline optimization and dataset-generation pipeline. Instances 1–10 are used during model development and policy selection, whereas instances 11–12 are kept separate for final evaluation.

Chapter 8

Computational Results

8.1 Offline Scheduling and Oracle Results

This section reports the main computational results obtained from the offline stages described above. The aim is not to repeat all intermediate outputs of the optimization pipeline, but to highlight the results that are relevant for the subsequent learning problem. The analysis first summarizes the schedules generated by the initial SAA_N procedure. It then examines the effect of full-case-mix scenario reduction, before considering the final setting with non-elective demand. The section concludes with the composition of the chronological ML-safe datasets. Unless otherwise stated, optimization results in this section refer to case-mix instances 1–10. Instances 11–12 are excluded from model and policy development and are retained for the final evaluation.

Initial SAA_N schedules. The initial SAA_N procedure produces one elective schedule for each instance, specialty, and scheduling strategy. Table 8.1 summarizes the performance of the selected candidates for instances 1–10 evaluated on the complete pool of 70 elective duration scenarios. In the table, OT/OR represents overtime per operating room, expressed in minutes.

Table 8.1: Performance of the initial SAA_N schedules for instances 1–10. Results are averaged over the ten case-mix instances within each strategy–specialty combination. Mean overtime is reported per operating room.

Strategy	Specialty	Scheduled patients	Served priority	Utilization (%)	Mean OT/OR (min)	Cancel rate (%)
Facility	General	68.81	39.83	82.63	7.75	13.00
Facility	Gyn & Obstetrics	88.66	51.66	78.50	11.53	12.13
Facility	Otolaryngology	138.11	70.89	98.32	7.33	5.78
Patient	General	79.69	61.03	83.99	10.70	13.47
Patient	Gyn & Obstetrics	101.19	72.60	79.14	16.12	13.89
Patient	Otolaryngology	144.16	88.45	99.73	8.73	5.28

The results reflect the different priorities of the two scheduling strategies. Across all three specialties, the patient-centered strategy serves more patients and achieves a substantially larger cumulative priority than the corresponding facility-centered strategy. This increase in patient-oriented performance is generally accompanied by a larger overtime requirement. In particular, mean overtime per operating room increases from 7.75 to 10.70 minutes for General Surgery and from 11.53 to 16.12 minutes for Gynecology & Obstetrics. A smaller increase is observed for Otolaryngology.

Scenario reduction and reduced-SAA evaluation. A two-dimensional visualization of the clustering obtained for one example case-mix is provided in Figure 6. We fit K-means in the original high-dimensional scenario space, but used the PCA projection for visualization.

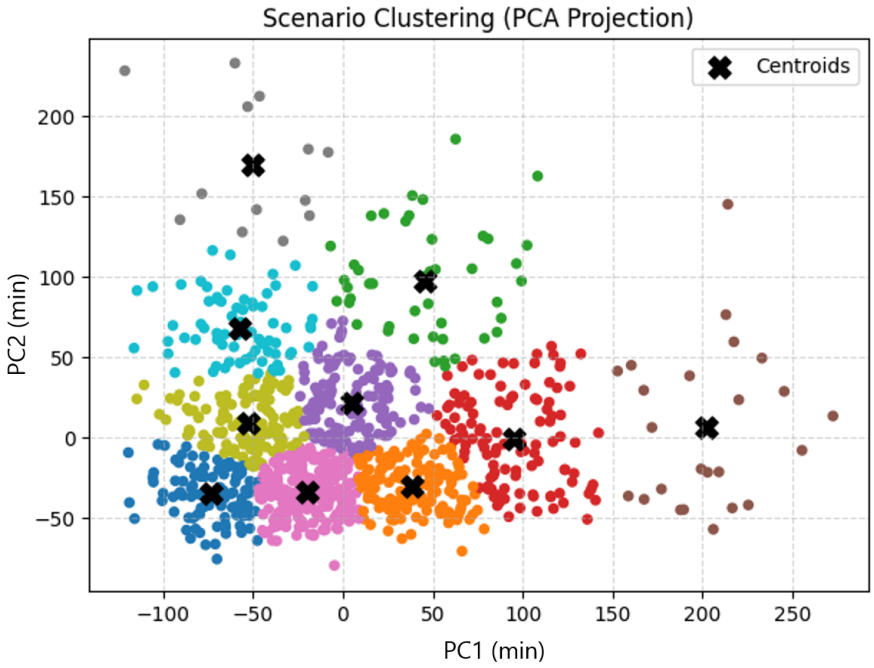

Figure 6: PCA projection of the 1000 full-case-mix duration scenarios generated for case-mix instance 1, General Surgery, under the facility-centered strategy. Each point represents one Monte Carlo realization of the complete case-mix. Colors identify the ten clusters obtained through K-means, while black crosses denote the projected cluster centroids.

We solved the weighted reduced-SAA problem independently for all 60 strategy-specific configurations associated with instances 1–10. In every configuration, the reduced-SAA solution selects exactly the same elective patient set as the corresponding original SAA_N solution. Hence, $N_{\mathrm{recovered}} = 0$ and $N_{\mathrm{dropped}} = 0$ over the complete set of development instances.

The out-of-sample evaluation then confirmed that, under the considered parameter setting, scenario reduction does not change the first-stage patient-selection decision. Consequently, the reduced formulation does not improve the patient-selection outcome relative to the initial SAA_N procedure; however, its advantage is instead computational. Based on the execution times recorded during the experiments, the five fold-specific optimization problems of the initial SAA_N procedure required approximately 40 minutes in total per configuration, whereas the weighted reduced-SAA problem required approximately 10 minutes per configuration.

The scenario-reduction preprocessing stage, including the clustering of the 1000 scenarios generated for each instance–specialty pair, required approximately 8 minutes in total. This stage was performed only once before solving the 60 configuration-specific reduced-SAA problems, and its computational cost was therefore shared across all configurations. Based on these indicative execution times, solving all 60 configurations required approximately 40 hours with the initial SAA_N procedure, compared with slightly more than 10 hours for the reduced-SAA procedure, including the one-time clustering stage. The observed computational time was therefore reduced by approximately 75%, while the same first-stage patient-selection decisions were preserved.

Non-elective scheduling and oracle results. Introducing non-elective demand has a visible effect on the number of elective patients selected by the reduced-SAA model. In the elective-only setting, the schedules for instances 1–10 contain on average 108.87 elective patients under the facility-centered strategy and 120.60 under the patient-centered strategy. After non-elective capacity pressure is included, these averages decrease to 91.57 and 101.53 patients, respectively instead. Hence, the reduction is approximately 16% under both strategies.

Table 8.2 reports the corresponding perfect-information oracle results over the original 1000 scenarios. The values are aggregated over instances 1–10.

Table 8.2: Perfect-information oracle results with non-elective demand, averaged over instances 1–10 and the original Monte Carlo scenarios.

Strategy	Scheduled	Executed	Canceled	Cancel rate (%)	Total OT (min)	$P(\mathrm{OT} > 0)$ (%)	NE patients	NE workload (min)
Facility	91.57	82.56	9.01	10.72	187.32	99.96	20.26	1143.17
Patient	101.53	92.15	9.38	9.98	178.08	99.98	20.26	1143.17

We observed that the feasibility slack introduced to accommodate extreme non-elective demand remains small in magnitude. The additional overtime has a mean value of about 0.69 minutes per scenario under both strategies and is positive in about 23.2% of facility-centered scenarios and 30.1% of patient-centered scenarios. In these 60,000 strategy-specific scenario evaluations, all non-elective patients are operated on.

Final supervised datasets. The chronological reconstruction described in Section 5.1 produces the datasets used by the classification models. Only states marked as reachable

are included in the ML-safe version. Table 8.3 reports the label distribution according to the instance-level split used in the subsequent experiments.

Table 8.3: Label distribution in the chronological reachable ML-safe datasets.

Instance subset	Cancel	Execute	Observations	Cancel rate (%)
1–6	185,826	3,132,992	3,318,818	5.60
7–8	68,116	1,068,857	1,136,973	5.99
9–10	62,770	1,039,432	1,102,202	5.69
11–12	62,450	1,038,644	1,101,094	5.67

Instances 1–10 contain a total of $5,557,993$ reachable decision states, of which 316,712 correspond to oracle cancellations and 5,241,281 to executions. General Surgery consistently exhibits the largest cancellation rate across subsets, followed by Gynecology & Obstetrics, while Otolaryngology shows substantially fewer cancellations.

8.2 Learning and Policy-Selection Results

Predictive comparison. Table 8.4 summarizes the best results obtained on instances 7–8 for the main predictive criteria using an execution threshold of $\tau = 0.50$.

Table 8.4: Best predictive results we obtained on instances 7–8 across all the 29 classifier configurations. The execution threshold used for binary metrics in this preliminary comparison is $\tau = 0.50$.

Metric	Best model configuration	Value
Accuracy	Random Forest, 200 trees, depth 18, leaf 100	0.8370
Balanced accuracy	XGBoost, depth 5, lr 0.05, 400 trees	0.8344
Cancellation F1-score	Random Forest, 200 trees, depth 18, leaf 100	0.3754
ROC-AUC	XGBoost, depth 5, lr 0.05, 400 trees	0.9100
Cancellation average precision	XGBoost, depth 5, lr 0.05, 400 trees	0.4263

Ensemble models achieve higher discrimination on the validation instances, particularly for the minority cancellation class. This suggests that, although the experiment does not isolate non-linearity as the sole explanation for the performance difference, additional model flexibility is useful for the state representation considered here.

Threshold behavior and joint policy selection. For the XGBoost configuration with maximum depth 5, learning rate 0.05, and 400 trees, the predictive metrics vary substantially with the execution threshold. Figure 7 shows the corresponding accuracy and cancellation F1-score over the examined operating points.

Figure 7 illustrates the different effects of the execution threshold on the two predictive criteria. Accuracy, in fact, decreases as the threshold becomes more conservative,

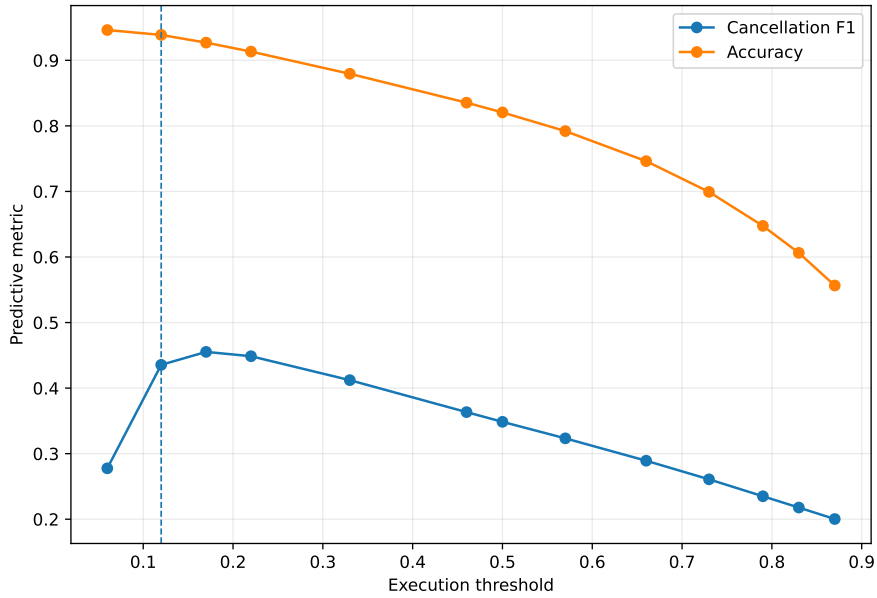

Figure 7: Accuracy and cancellation F1-score across the examined execution thresholds for the XGBoost configuration with maximum depth = 5, learning rate = 0.05, and 400 trees. The vertical dashed line indicates the selected execution threshold $\tau^\star = 0.12$.

whereas cancellation F1-score initially increases and reaches its highest value at a threshold slightly above the selected operating point before declining. Hence, neither predictive metric alone determines the final threshold.

Figure 8 shows that the threshold ultimately selected for this XGBoost configuration is not the optimum of a single predictive criterion. The minimum of the joint criterion occurs at $\tau = 0.12$, reflecting the intended balance between operational and predictive performance.

Threshold and selected policy. The joint validation procedure selects the XGBoost configuration with maximum depth 5, learning rate 0.05, 400 boosting trees, and an optimal execution threshold of $\tau^\star = 0.12$. On instances 9–10, this policy produces a predicted cancellation rate of 5.16% (close to the oracle rate of 5.69%), with an accuracy of 0.9387, cancellation F1-score of 0.4355, balanced accuracy of 0.6927, and ROC-AUC of 0.9082.

The operational-validation results of the selected candidate on 9–10 are reported in Table 8.5.

The selected model is not simply the classifier with the highest value of one predictive metric. Its final normalized selection score is approximately 0.0835, combining the operational rank with threshold-specific cancellation F1-score and accuracy. This emphasizes the distinction between selecting a statistical classifier and selecting an online decision policy.

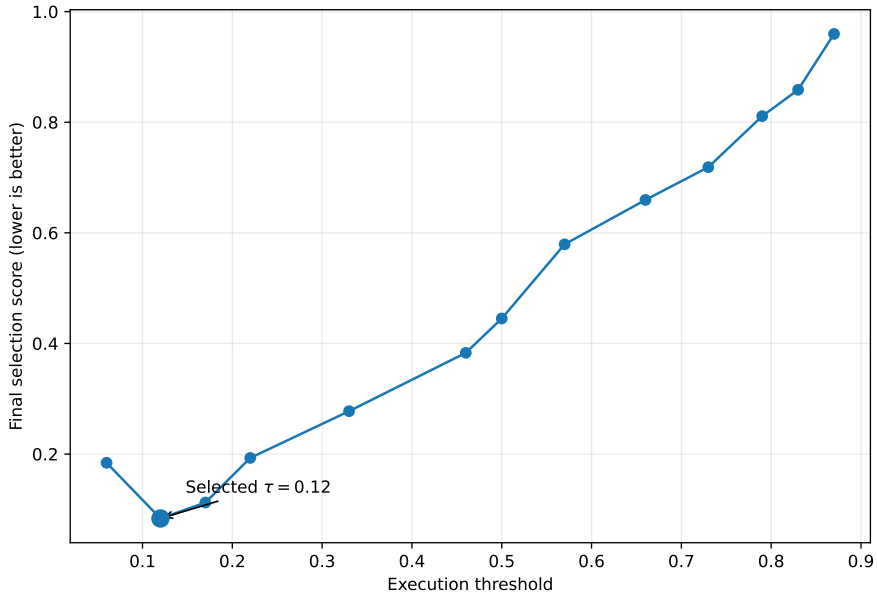

Figure 8: Joint policy-selection score across the candidate execution thresholds of the XGBoost configuration with depth = 5, learning rate = 0.05, and 400 trees. The score components are based on ranks computed over the complete candidate-policy set; lower scores are preferred. The minimum occurs at the selected threshold $\tau^\star = 0.12$.

Table 8.5: Selected XGBoost policy and its dynamic operational-validation results on instances 9–10.

Measure	Patient strategy	Facility strategy
Executed elective priority	68.31	44.67
Executed elective duration (min)	3919.74	3929.82
Canceled elective patients	7.56	7.58
Regular overtime (min)	236.43	226.27
Extra overtime (min)	267.10	222.05
Total overtime (min)	503.53	448.32
Unserved non-elective patients	0.00	0.00
Mean non-elective waiting time (min)	14.91	17.21
Non-elective within-limit rate (%)	97.69	96.76

Sensitivity to policy-selection weights. A local sensitivity analysis was performed to assess whether the selected policy depends critically on the baseline weighting

$$(w_{\text{op}}, w_{F_1}, w_{\text{acc}}) = (0.50, 0.25, 0.25).$$

Table 8.6 reports the policy selected under the seven weighting specifications introduced in the experimental setup.

Table 8.6: Sensitivity of policy selection to local changes in the weights of the joint selection criterion.

w_{op}	w_{F_1}	w_{acc}	Model	Selected configuration	τ
0.45	0.25	0.30	XGBoost	depth 5, lr 0.05, 400 trees	0.12
0.45	0.30	0.25	XGBoost	depth 5, lr 0.05, 400 trees	0.12
0.50	0.20	0.30	XGBoost	depth 5, lr 0.03, 400 trees	0.12
0.50	0.25	0.25	XGBoost	depth 5, lr 0.05, 400 trees	0.12
0.50	0.30	0.20	XGBoost	depth 5, lr 0.05, 400 trees	0.12
0.55	0.20	0.25	XGBoost	depth 5, lr 0.03, 400 trees	0.12
0.55	0.25	0.20	XGBoost	depth 5, lr 0.05, 400 trees	0.12

The baseline policy remains the top-ranked candidate under five of the seven weighting specifications and ranks second under the remaining two. Across all seven specifications, the selected model family, maximum depth, number of boosting trees, and execution threshold remain unchanged. The only variation occurs for the two specifications $(0.50, 0.20, 0.30)$ and $(0.55, 0.20, 0.25)$, for which the selected XGBoost learning rate changes from 0.05 to 0.03, while the execution threshold remains fixed at $\tau = 0.12$.

These results indicate that the selected operating point is locally stable with respect to moderate changes in the relative weights of the three components of the selection criterion. In particular, the selection of XGBoost with depth 5, 400 trees, and threshold $\tau = 0.12$ does not depend on the exact baseline weighting.

Static re-evaluation on instances 11–12. The refitted classifier is then evaluated on the corrected chronological datasets associated with instances 11–12. These observations constitute the corrected chronological states used for the static re-evaluation on instances 11–12. The experiment measures predictive agreement with the oracle on a fixed collection of decision states, but it is not yet a closed-loop operational evaluation of the learned policy. Table 8.7 reports the aggregate results.

The class imbalance makes overall accuracy alone a weak description of the cancellation problem. Among the 62,450 oracle cancellation states, 25,384 are correctly identified as cancellations, whereas 37,066 are classified as executions. Among the 1,038,644 oracle execution states, 30,252 are classified as cancellations and 1,008,392 are correctly executed. The two error types have different operational meanings: an oracle cancellation classified as execution preserves elective activity but consumes capacity that the ora-

Table 8.7: Corrected static re-evaluation of the locked policy on instances 11–12 (1,101,094 total observations).

Metric	Value
Accuracy	0.9389
Balanced accuracy	0.6887
Cancellation precision	0.4563
Cancellation recall	0.4065
Cancellation F1-score	0.4299
ROC-AUC	0.9084
Cancellation average precision	0.4101
True cancellation rate	5.67%
Predicted cancellation rate	5.05%

cle would have protected, whereas an oracle execution classified as cancellation removes elective activity that the oracle would have retained.

The balanced accuracy of 0.6887, cancellation F1-score of 0.4299, and ROC-AUC of 0.9084 therefore provide a more informative view of the final classifier than accuracy alone.

The predicted cancellation frequency, 5.05%, is slightly lower than the oracle cancellation frequency, 5.67%. Hence, at the locked threshold the classifier is moderately more execution-oriented than the oracle on these states.

Performance is not completely homogeneous across specialties. Across the 12 instance–strategy–specialty groups, cancellation F1-score ranges approximately from 0.306 to 0.496. The lowest values occur in some Otolaryngology groups, where the underlying cancellation prevalence is also considerably lower than in General Surgery and Gynecology and Obstetrics. This confirms the importance of complementing aggregate classification metrics with the closed-loop analysis reported in the following section.

Finally, the 11–12 static re-evaluation does not reproduce the state trajectory that would be generated by the learned policy itself. If the classifier makes a different decision from the oracle, subsequent states may also change. The final assessment must therefore be performed in a closed-loop environment. The results reported in Section 8.3 address this issue by evaluating the locked model and threshold in the discrete-event simulation.

Post-hoc interpretation of the final classifier. The internal structure of the locked XGBoost classifier was examined post hoc using total-gain feature importance. For categorical predictors, the total gain of the corresponding one-hot encoded levels was aggregated back to the original variable. This analysis was performed only after the model configuration and execution threshold had been fixed and therefore played no role in policy selection.

Figure 9 shows that the fitted classifier relies primarily on variables describing the relationship between the current elective case and the residual operating-room workload.

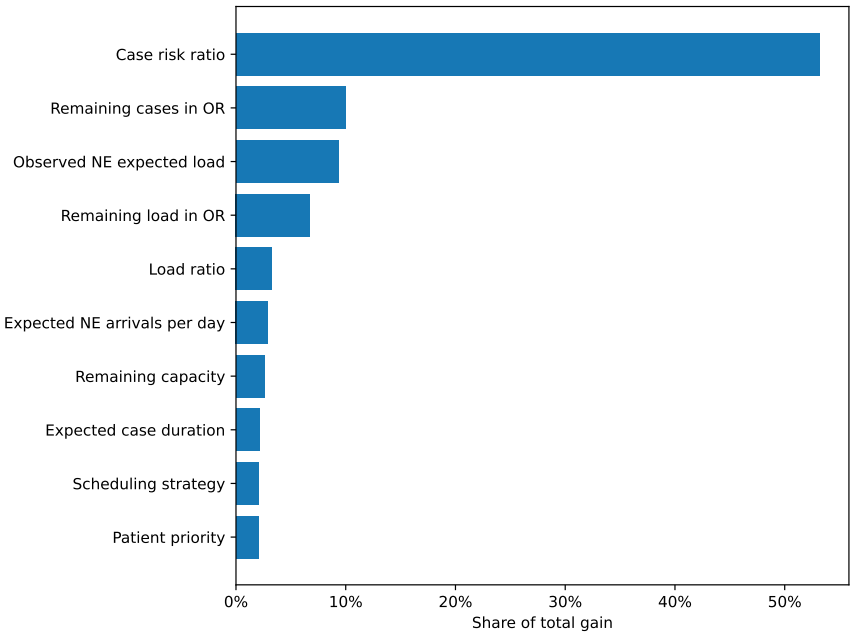

Figure 9: Total-gain feature importance of the final XGBoost classifier. One-hot encoded categorical levels are aggregated to the corresponding original online-safe predictor. The analysis is post-hoc and does not enter model or threshold selection.

`case_risk_ratio` accounts for approximately 53.2% of the total gain, followed by the number of remaining cases in the operating room (10.0%), the expected non-elective load already observed during the day (9.3%), and the remaining operating-room workload (6.7%). Remaining capacity, expected case duration, scheduling strategy, and patient priority each make smaller but still visible contributions.

The concentration of importance on workload- and capacity-related variables is consistent with the operational nature of the execute/cancel decision. At the same time, these values describe the split structure of the fitted tree ensemble rather than causal effects. They do not indicate the direction of each relationship, and importance may also be distributed unevenly among correlated predictors.

8.3 Closed-Loop Simulation Results

Aggregate policy comparison. Tables 8.8 and 8.9 report the main results of the final 1000-replication experiment. Each entry is averaged over $2 \times 3 \times 1000 = 6000$ weekly trajectories for the corresponding policy and scheduling strategy.

Figures 10 and 11 make the policy trade-off visible in the two objective spaces used throughout the thesis.

The always-execute rule preserves the largest amount of elective activity, as expected, but produces by far the largest extra overtime. Relative to this benchmark, the

Table 8.8: Final DES results under the facility-centered schedules. Overtime measures are weekly totals across operating-room sessions.

Policy	Canceled patients	Executed duration (min)	Regular OT (min)	Extra OT (min)	NE mean wait (min)	NE within limit (%)
Selected XGBoost $\tau = 0.12$	7.01	3956.63	235.36	218.12	16.32	97.29%
Always execute	0.00	4256.90	266.45	481.39	16.49	97.26%
Expected-capacity rule	11.14	3731.82	151.41	85.55	16.16	97.32%
XGBoost $\tau = 0.50$	25.21	3117.07	94.51	71.40	13.09	97.80%

Table 8.9: Final DES results under the patient-centered schedules. Overtime measures are weekly totals across operating-room sessions.

Policy	Canceled patients	Executed priority	Regular OT (min)	Extra OT (min)	NE mean wait (min)	NE within limit (%)
Selected XGBoost $\tau = 0.12$	7.59	67.14	235.62	255.15	15.07	97.70
Always execute	0.00	70.77	271.92	497.87	15.34	97.66
Expected-capacity rule	13.25	63.65	154.68	80.94	15.18	97.69
XGBoost $\tau = 0.50$	24.41	57.96	116.13	83.56	12.58	98.07

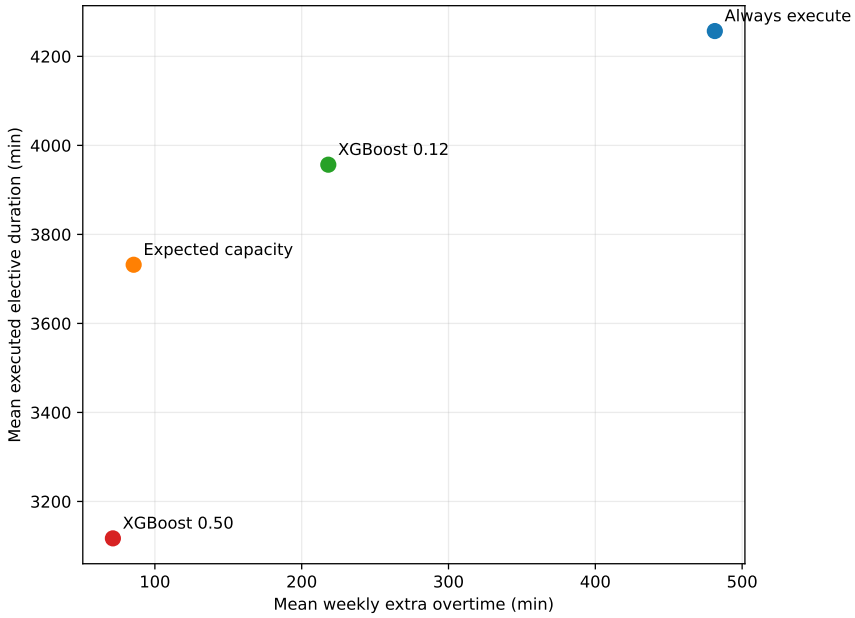

Figure 10: Trade-off between executed elective workload and extra overtime under the facility-centered schedules. Each point represents one online decision policy averaged over the final DES replications.

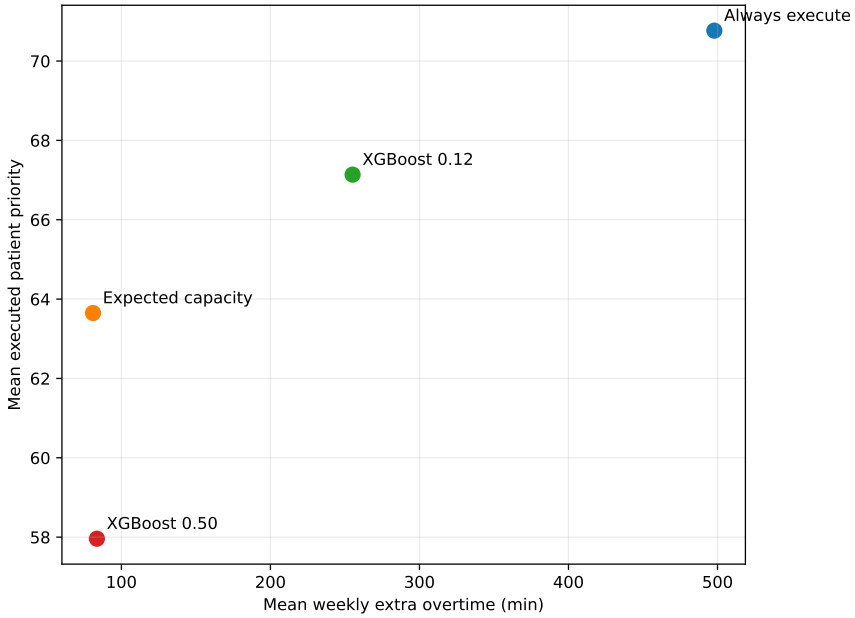

Figure 11: Trade-off between executed patient priority and extra overtime under the patient-centered schedules. Each point represents one online decision policy averaged over the final DES replications.

selected policy reduces mean extra overtime by approximately 263.27 minutes under the facility-centered strategy and 242.72 minutes under the patient-centered strategy. These reductions correspond to approximately 55% and 49%, respectively.

The expected-capacity rule is substantially more conservative. Compared with this rule, the selected policy executes an additional 224.81 minutes of elective activity under the facility-centered schedules and obtains approximately 3.49 additional units of executed patient priority under the patient-centered schedules. It also produces approximately 4.14 and 5.66 fewer elective cancellations, respectively. The price of this higher elective service level is larger extra overtime.

Paired comparisons. Paired comparisons confirm these tradeoffs with narrow 95% confidence intervals. Furthermore, approximately 24% to 25% of all cancellations observed under the selected policy are indirect prefix cancellations triggered when removing an initial canceled case from an operating room sequence.

Direct and prefix cancellations. An elective cancellation produced directly by the online policy may generate additional cancellations through the prefix rule. We distinguish between *direct cancellations*, corresponding to patients for which the policy explicitly returns a cancel decision, and *prefix cancellations*, corresponding to later patients removed from the same operating-room sequence.

Table 8.10: Paired comparisons between the selected policy and the main benchmarks. Values in brackets are 95% confidence intervals. For the facility-centered strategy, the primary outcome is executed elective workload (min); for the patient-centered strategy, it is executed patient priority (priority units).

Strategy	Comparator	Δ primary outcome	Δ cancellations	Δ extra OT (min)
Facility	Always execute	−300.28 [−305.17, −295.38]	+7.01 [6.90, 7.11]	−263.27 [−267.93, −258.61]
Facility	Expected capacity	+224.81 [221.46, 228.15]	−4.14 [−4.20, −4.08]	+132.58 [130.33, 134.83]
Patient	Always execute	−3.63 [−3.69, −3.57]	+7.59 [7.47, 7.71]	−242.72 [−247.30, −238.14]
Patient	Expected capacity	+3.49 [3.43, 3.55]	−5.66 [−5.76, −5.57]	+174.20 [171.33, 177.08]

Table 8.11 reports this decomposition for the selected policy.

Table 8.11: Decomposition of elective cancellations under the selected XGBoost policy.

Strategy	Total	Direct	Prefix	Prefix share
Facility	7.01	5.25	1.75	25.0%
Patient	7.59	5.77	1.82	24.0%

Thus, approximately one quarter of all cancellations observed under the selected policy are not direct classifier decisions but consequences of the sequential prefix constraint. The mean direct cancellation rates are approximately 6.18% and 6.29% for the facility-centered and patient-centered schedules, respectively, while prefix propagation adds approximately two further percentage points.

This distinction is important when comparing the DES with the static classification results reported above. A classification-level predicted cancellation rate cannot be interpreted directly as the final proportion of elective patients removed from the executed schedule, because a single online cancellation may affect multiple downstream elective cases.

Static-to-dynamic state shift. The static 11–12 re-evaluation reported above produced a predicted cancellation rate of approximately 5.05%. The dynamic DES produces a larger frequency of direct low-score decisions. This difference does not imply that the classifier or its threshold has changed. Instead, the learned policy changes the distribution of states on which it is subsequently queried.

We then performed a post-hoc diagnostic comparison between the XGBoost execution scores obtained on the static final re-evaluation states and those observed at states actually visited by the selected policy inside the DES. The main results are reported in Table 8.12.

The shift is moderate in absolute terms, but it occurs in the direction expected from the closed-loop setting: states visited during actual simulation receive slightly lower execution scores and are more likely to fall below the selected threshold.

This result provides an empirical explanation for part of the difference between the static and dynamic evaluations. Non-elective arrivals and previous elective decisions

Table 8.12: Execution-score distribution in the static re-evaluation and along trajectories visited by the learned policy.

State source	Observations	Mean score	Share below 0.12
Static final re-evaluation states	1,101,094	0.7256	5.05%
DES visited states	225,306	0.7151	5.92%

modify operating-room clocks, residual capacity, remaining workload, and other online predictors. These changes are then reflected in the score assigned to later elective patients.

The remaining difference between direct decision rates and total cancellation rates is explained by the prefix mechanism discussed above. Hence, the final cancellation behavior is generated by two distinct effects: endogenous movement of the classifier through the state space and propagation of individual cancel decisions to later patients in the same operating-room sequence.

Post-hoc threshold sensitivity. The final threshold $\tau^\star = 0.12$ is fixed before instances 11–12 are evaluated and is not re-optimized using the final simulation results. Nevertheless, a limited post-hoc sensitivity analysis is useful for checking whether the observed trade-off behaves smoothly in a neighborhood of the selected operating point.

For diagnostic purposes only, the refitted XGBoost model was simulated at thresholds

$$\tau \in \{0.08, 0.10, 0.12, 0.14\}$$

using 100 replications for each instance–specialty–strategy combination.

The results exhibit the expected monotone trade-off. Under the facility-centered strategy, increasing the threshold from 0.08 to 0.14 increases mean elective cancellations from approximately 4.35 to 8.16, while reducing extra overtime from approximately 303.14 to 192.44 minutes. Under the patient-centered strategy, cancellations increase from approximately 4.82 to 8.78, whereas extra overtime decreases from approximately 329.16 to 230.98 minutes.

The selected threshold 0.12 lies inside this smooth local trade-off rather than representing an isolated numerical outcome. However, these post-hoc results are purely diagnostic and are not used to reconsider the policy selected on instances 9–10. In particular, the explored threshold range is not wide enough to establish an exact equal-cancellation or equal-overtime comparison with the expected-capacity rule.

Chapter 9

Conclusions and Future Research

The central problem addressed in this thesis is the mismatch between the time at which a detailed operating-room decision can be optimized and the time at which it must be implemented. Before execution, uncertainty can be represented through scenarios, and relatively rich optimization models can be solved. During execution, decisions must instead react to the current state of the system as durations and non-elective demand are progressively revealed.

The framework developed here connects these two settings. Stochastic optimization is used offline to construct weekly schedules and generate high-quality execute/cancel decisions. Supervised learning transfers those decisions to a state-to-action mapping based only on information available online. Closed-loop simulation then evaluates the consequences of allowing that learned mapping to control the operating-room trajectory.

9.1 Main Findings and Contributions

The offline experiments show that uncertainty can be incorporated before real-time execution without fixing a single deterministic view of surgical duration. Scenario reduction provides a compact representation of the complete case mix, and in the development experiments the weighted reduced-SAA solutions preserve the same elective patient set as the corresponding initial stochastic schedules. Introducing stochastic non-elective workload then places those schedules under the shared capacity pressure that motivates the later online decision problem.

The methodological transition from optimization to learning occurs at the perfect-information oracle. This comes from the fact that the oracle can use the complete realized scenario to identify desirable execute/cancel actions, but those same inputs cannot be given to an implementable classifier. The chronological state reconstruction consequently separates the information used to define the target from the information used to predict it. This separation is fundamental because it prevents future-information leakage and defines the online decision problem actually studied by the classifier.

The classification experiments also show that predictive model selection and policy selection are different tasks. Ensemble methods provide the strongest discrimination on

the validation instances, but the conventional threshold 0.50 produces behavior that is poorly aligned with the observed cancellation prevalence. Model configuration and execution threshold are therefore selected jointly using predictive and operational evidence. With this procedure, we selected an XGBoost model with depth 5, learning rate 0.05, 400 trees, and threshold $\tau^\star = 0.12$.

In the closed-loop DES, we observed that "always execute" preserves the largest amount of elective activity, but this comes with substantially more extra overtime; in contrast, the "expected-capacity" rule controls overtime more aggressively, at the cost of additional cancellations and lower elective performance. This means that the XGBoost policy sacrifices some elective performance relative to always execute, while retaining more executed duration in facility-centered schedules and more patient priority in patient-centered schedules than the expected-capacity rule.

The dynamic experiment also shows why the static classification results provide only part of the picture: once the classifier is used to make execute/cancel decisions, those decisions modify the remaining workload and the timing of later decision points. The states visited in the DES are, therefore, not exactly the same as those contained in the oracle-guided static evaluation dataset. Approximately one quarter of the elective cancellations under the selected policy are also generated indirectly by prefix propagation rather than by a direct classifier decision. One predicted action can affect several subsequent procedures accordingly.

The main contribution of the thesis lies in this offline-to-online connection. Optimization provides decision knowledge that would be too expensive or too informed to reproduce directly at every online decision point; supervised learning provides a fast approximation based on the observable state; and simulation reveals whether that approximation remains useful once its actions alter the system it controls. The same logic is not specific to operating rooms and can apply more broadly to sequential decision problems in which high-quality offline optimization is available but repeated real-time optimization is impractical.

9.2 Limitations

The scope of the results is determined in part by the empirical basis of the experiments. The processed benchmarking data do not represent the complete operating environment of a specific hospital. The processed benchmarking data combine observations from multiple institutions and organizational settings, while the experiments focus on the Basic and Regular Care level and on three surgical specialties. The 12 case-mix instances considered for each specialty are synthetic populations generated from the available empirical information rather than observed weekly waiting lists from a single hospital.

Several patient and system characteristics are also modeled synthetically. In particular, elective patient priorities are experimental weights rather than values derived from a validated clinical urgency scale. Similarly, the proportion and arrival process of non-elective patients are modeling assumptions and are not calibrated to hospital-specific emergency-arrival data.

The optimization models focus on operating-room capacity and leave out several resources that may become binding in practice. In particular, the availability of surgeons, anesthesiologists, nursing staff, equipment, post-anesthesia care units, intensive-care beds, and downstream wards is not explicitly represented. Adding these constraints could affect both the schedules generated offline and the execute/cancel decisions that are preferred during execution.

It is clear that the oracle is a simplified representation of the decision environment, and not a complete model of hospital operations. We use it to provide informative execute/cancel labels under complete scenario information. Indeed, non-elective demand is handled at an aggregate daily level in the oracle, while the final discrete-event simulation represents individual arrivals in chronological order. They are two components that describe the same operating-room system at different levels of detail.

The action space is another simplification. The learned policy decides whether the next elective patient should be executed or canceled. Once a cancellation occurs, the current case and the remaining elective suffix of that operating-room sequence are removed, so that the executed elective cases form a prefix of the original sequence. A real operating-room manager may instead be able to resequence cases, move a procedure to another room, delay it temporarily, exchange patients between sessions, modify staffing, or explicitly decide how much overtime to authorize.

The machine-learning model is trained on states generated by the offline optimization pipeline. Even though the chronological reconstruction removes future information from the predictors, most of the training data still come from oracle-guided decisions. When the learned policy is tested in the discrete-event simulation, its independent choices modify subsequent states in the trajectory as a result. So during the DES the classifier deals with a group of states that is a little different from the ones it was trained on, and the current way of training does not directly fix this difference.

The final evaluation is based on simulation rather than on data from an independent hospital. Given that instances 11–12 are not used for model fitting, hyperparameter selection, or threshold selection, and common random numbers are used so that the alternative policies are compared under the same stochastic realizations, this provides a controlled experimental comparison, but it does not constitute external validation. The reported results, in turn, should be interpreted as evidence on the behavior of the proposed method within the stochastic environment modeled in this study, rather than as a direct estimate of its effect in clinical practice.

In addition, all non-elective patients are eventually served by design in the final simulation. This ensures feasibility and makes waiting times, deadline violations, and overtime the relevant indicators of non-elective performance, but it prevents the number of unserved non-elective patients from acting as a discriminating performance measure among the simulated policies.

9.3 Directions for Future Research

Empirical calibration and external validation are the most immediate extensions. Access to hospital-specific elective waiting lists, non-elective arrival records, staffing information, and clinically validated priority measures would make it possible to replace several experimental assumptions with institution-specific estimates. The complete framework could then be evaluated on schedules observed in practice and, before prospective deployment, compared with decisions made by experienced operating-room managers.

The stochastic duration model could also be enriched. Rather than sampling different peri-operative phases independently, future work could represent dependence between phases and introduce covariates related to patient characteristics, surgical teams, procedure complexity, and operating-room conditions. Non-elective arrivals could similarly be modeled through non-homogeneous or specialty-dependent processes when suitable empirical data are available.

The current formulation treats operating-room time as the main capacity constraint, but more detailed model could also account for the availability of surgeons, anesthesiologists, nursing teams, recovery beds, intensive-care capacity, equipment, and downstream wards. With these additional constraints, online decisions would reflect congestion across the surgical pathway rather than only the remaining operating-room capacity..

The decision itself could also be made less restrictive. In fact, here, the policy has only two possible actions: execute or cancel the next elective procedure, but future work could allow the policy to postpone a case, modify the remaining sequence, reassign a patient to another operating room, or authorize additional overtime. Such a more flexible decision-support system would, however, require a substantially larger optimization-generated action space and a more complex learning problem.

A particularly relevant methodological direction is to address the difference between oracle-guided training states and policy-induced deployment states. The present DES shows that this difference is observable even when the model is trained using online-safe features. Accordingly, future work could adopt iterative imitation-learning approaches in which the current learned policy is allowed to generate new trajectories and the optimization oracle is queried again on the states that the policy actually visits.

An alternative would be to use reinforcement-learning or approximate dynamic-programming approaches, which could optimize long-run operational performance rather than imitate individual oracle decisions. Such methods would require careful reward design and considerably stronger validation, but they could naturally account for the effect of one decision on later states.

The relationship between predictive uncertainty and operational decisions also deserves further investigation. The present framework uses a fixed execution threshold after policy selection. Future work could consider calibrated predictive probabilities, uncertainty-aware decision rules, or thresholds that vary according to specialty, current capacity pressure, or the observable non-elective state. Any such adaptation would need to be selected on development data and evaluated on a separate final sample in order to preserve the experimental separation used in this thesis.

Further research could extend the post-hoc interpretability analysis beyond global total-gain importance. Feature-attribution methods such as SHAP values, local explanations, or simplified surrogate rules could be used to investigate both the direction and the case-specific contribution of residual capacity, patient priority, remaining workload, and observed urgent demand to the model's recommendations. In a healthcare application, this would be important not only for methodological understanding but also for communicating individual recommendations to clinical and managerial users.

The current experiments stop at the end of a single simulated week. A longer planning horizon would allow canceled patients to re-enter the elective waiting list and affect the schedules constructed for subsequent weeks. The consequences of a cancellation could then be measured in terms of both immediate elective activity and the additional delay experienced by the patient. This would also make it possible to compare online policies according to their cumulative effects over several scheduling periods, rather than only within the week in which the decision is made.

9.4 Final Remarks

Operating-room scheduling under uncertainty requires decisions at two different timescales. Schedules must first be constructed without knowing the exact durations and urgent demand that will occur, and they must then be adapted as this uncertainty is progressively revealed. Treating these two stages independently risks producing either computationally detailed offline models that are difficult to use in real time or simple online rules that ignore the structure of the underlying optimization problem.

The framework developed in this thesis offers an intermediate approach. Optimization is used where computational time and complete scenario information can be exploited, while machine learning transfers part of the resulting decision structure to an online rule based only on observable information. Simulation then provides a controlled environment for testing whether this transfer remains operationally meaningful when the learned policy determines its own trajectory.

The computational results do not establish a universally optimal policy, nor do they eliminate the need for hospital-specific calibration and prospective validation. They do, however, show that optimization-derived decisions can be transformed into an adaptive supervised-learning policy that produces a meaningful trade-off between elective service and overtime under stochastic demand.

In this sense, the main result of the thesis is methodological. Rather than viewing optimization, machine learning, and simulation as competing tools, the proposed framework assigns a complementary role to each of them: optimization provides the expert decision structure, supervised learning makes that structure available at online decision times, and simulation evaluates its consequences under uncertainty.

Bibliography

[ABK25] M. Al Amin, R. Baldacci, and V. Kayvanfar. "A Comprehensive Review on Operating Room Scheduling and Optimization". In: *Operational Research* 25, 3 (2025). DOI: 10.1007/s12351-024-00884-z.

[BD07] J. Beliën and E. Demeulemeester. "Building Cyclic Master Surgery Schedules with Leveled Resulting Bed Occupancy". In: *European Journal of Operational Research* 176.2 (2007), pp. 1185–1204. DOI: 10.1016/j.ejor.2005.06.063.

[Bel+24] V. Bellini et al. "Artificial Intelligence in Operating Room Management". In: *Journal of Medical Systems* 48.1, 19 (2024). DOI: 10.1007/s10916-024-02038-2.

[Ber+25] A. M. Bernardelli et al. "Multi-Objective Stochastic Scheduling of Inpatient and Outpatient Surgeries". In: *Flexible Services and Manufacturing Journal* 37.4 (2025), pp. 1148–1202. DOI: 10.1007/s10696-024-09542-0.

[Çal+26] Y. K. Çalışkan et al. "Beyond Block Time: A Head-to-Head Comparison of Reinforcement Learning, Genetic Algorithms, and Predict-Then-Optimize Scheduling for Operating Room Workflow Using Discrete-Event Simulation". In: *International Journal of Medical Informatics* 214, 106426 (2026). DOI: 10.1016/j.ijmedinf.2026.106426.

[CDB10] B. Cardoen, E. Demeulemeester, and J. Beliën. "Operating Room Planning and Scheduling: A Literature Review". In: *European Journal of Operational Research* 201.3 (2010), pp. 921–932. DOI: 10.1016/j.ejor.2009.04.011.

[DA18] D. Duma and R. Aringhieri. "The Real Time Management of Operating Rooms". In: *Operations Research Applications in Health Care Management*. Ed. by C. Kahraman and Y. I. Topcu. Vol. 262. International Series in Operations Research & Management Science. Springer, 2018, pp. 55–79. DOI: 10.1007/978-3-319-65455-3_3.

[DA19] D. Duma and R. Aringhieri. "The Management of Non-Elective Patients: Shared vs. Dedicated Policies". In: *Omega* 83 (2019), pp. 199–212. DOI: 10.1016/j.omega.2018.03.002.

[Dal+26] A. Daldossi et al. “Prediction Accuracy or Decision Quality? A Predict-Then-Optimize Perspective on Surgical Case Assignment and Sequencing”. In: *Shaping a Sustainable Future in the Era of Big Data. AIRO Springer Series*. AIRO Springer Series (2026), pp. 1–15.

[FB22] J. J. Forbus and D. Berleant. “Discrete-Event Simulation in Healthcare Settings: A Review”. In: *Modelling* 3.4 (2022), pp. 417–433. DOI: 10.3390/modelling3040027.

[Gon+25] X. Gong et al. “A Stochastic Optimization Model and Decomposition Techniques for Surgery Scheduling with Duration and Emergency Demand Uncertainty”. In: *Computers & Industrial Engineering* 204, 111096 (2025). DOI: 10.1016/j.cie.2025.111096.

[Hul+12] P. J. H. Hulshof et al. “Taxonomic Classification of Planning Decisions in Health Care: A Structured Review of the State of the Art in OR/MS”. In: *Health Systems* 1.2 (2012), pp. 129–175. DOI: 10.1057/hs.2012.18.

[JJS99] J. B. Jun, S. H. Jacobson, and J. R. Swisher. “Application of Discrete-Event Simulation in Health Care Clinics: A Survey”. In: *Journal of the Operational Research Society* 50.2 (1999), pp. 109–123. DOI: 10.1057/palgrave.jors.2600669.

[KZ24] G. Korzhenevich and A. Zander. “Leveraging the Potential of the German Operating Room Benchmarking Initiative for Planning: A Ready-to-Use Surgical Process Data Set”. In: *Health Care Management Science* 27.3 (2024), pp. 328–351. DOI: 10.1007/s10729-024-09672-9.

[Lex+25] J. R. Lex et al. “Using Machine Learning to Predict-Then-Optimize Elective Orthopedic Surgery Scheduling to Improve Operating Room Utilization: Retrospective Study”. In: *JMIR Medical Informatics* 13, e70857 (2025). DOI: 10.2196/70857.

[Liu24] H. Liu. “Model-Driven Solution Approaches for Patient Scheduling in Admission and Surgery Management”. PhD thesis. University of Angers, 2024.

[Man+24] J. Mandi et al. “Decision-Focused Learning: Foundations, State of the Art, Benchmark and Future Opportunities”. In: *Journal of Artificial Intelligence Research* 80 (2024), pp. 1623–1701. DOI: 10.1613/JAIR.1.15320.

[MF25] I. E. S. de Melo and F. S. Fogliatto. “Integration of Decision Levels in Operating Room Scheduling Problems: Systematic Review and Proposition of a Decision Support Framework”. In: *Computers & Operations Research* 180, 107063 (2025). DOI: 10.1016/j.cor.2025.107063.

[Oml+18] E. Omling et al. “Population-Based Incidence Rate of Inpatient and Outpatient Surgical Procedures in a High-Income Country”. In: *BJS* 105.1 (2018), pp. 86–95. DOI: 10.1002/bjs.10643.

[Pha+23] T.-S. Pham et al. “A Prediction-Based Approach for Online Dynamic Appointment Scheduling: A Case Study in Radiotherapy Treatment”. In: *INFORMS Journal on Computing* 35.4 (2023), pp. 844–868. DOI: 10.1287/ijoc.2023.1289.

[PWH24] Z. Pu, S. Wu, and Y. Han. “A Discrete-Event Simulation Model for Assessing Operating Room Efficiency of Thoracic, Gastrointestinal, and Orthopedic Surgeries”. In: *World Journal of Surgery* 48.5 (2024), pp. 1102–1110. DOI: 10.1002/wjs.12116.

[VD15] C. Van Riet and E. Demeulemeester. “Trade-offs in Operating Room Planning for Electives and Emergencies: A Review”. In: *Operations Research for Health Care* 7 (2015), pp. 52–69. DOI: 10.1016/j.orhc.2015.05.005.

[Wul+07] G. Wullink et al. “Closing Emergency Operating Rooms Improves Efficiency”. In: *Journal of Medical Systems* 31.6 (2007), pp. 543–546. DOI: 10.1007/s10916-007-9096-6.

Ringraziamenti

Prima di tutto vorrei ringraziare il Professor Duma per la disponibilità e il tempo che mi ha dedicato. La stesura di questo lavoro non sarebbe stata possibile senza la sua guida.

Un ringraziamento speciale va ai miei genitori per l'affettuoso sostegno che mi hanno dimostrato durante questo percorso accademico. È stato prezioso affrontare gli studi con la serenità che mi hanno donato. Hanno inoltre sempre ascoltato con pazienza quando capitava che io iniziassi dei discorsi da matematica in casa: spero di averli confusi quel tanto che basta.

Ringrazio poi mia sorella, perché ogni tappa della nostra vita fino ad ora è stata affrontata insieme. Non riesco a immaginare cosa sia la solitudine, e mi va bene così.

Vorrei ringraziare le mie amiche del liceo. Vedere che ognuna di noi ora sta costruendo la sua vita mi emoziona ogni volta che ci incontriamo.

Ringrazio inoltre tutti i miei colleghi matematici. Abbiamo scelto di studiare una disciplina tanto affascinante quanto complessa. Affrontare le sue difficoltà insieme ha reso ancora più stupendo questo percorso.