

Optimising Elective Theatre Lists: Machine Learning and Mathematical Optimisation for Patient Scheduling

Madhu Sudan Sapkota

madhu.sapkota07@outlook.com

ESNEFT, Colchester UK <https://orcid.org/0009-0002-6487-5749>

Michael Kampouridis

University of Essex, Colchester, UK

Hugo Herrera

ESNEFT, Colchester UK

Faiyaz Doctor

University of Essex, Colchester, UK

Xinan Yang

University of Essex, Colchester, UK

Research Article

Keywords: Surgery Scheduling, Meta-heuristic, Integer Linear Programming, Simulated-Annealing

Posted Date: March 17th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9140578/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

1 Introduction

The backlog of Referral-to-Treatment (RTT) patients awaiting surgery remains a critical challenge in many countries [1]. In the UK, despite the efficiency of the National Health Service (NHS), the number of patients waiting more than a year for surgery has been steadily increasing [2]. Addressing this issue is crucial and remains a top priority for healthcare sectors, including the NHS. However, this is a challenging task, as planning surgeries and managing associated resources is a complex process—uncertain timing, frequently constrained, and continuing to consume a significant portion of total hospital costs [3].

Surgical scheduling and planning can be divided into three levels of decision-making: strategic, tactical, and operational [4]. The strategic level involves long-term decisions, such as capacity planning to maximise profit or cost efficiency. This level often focuses on the “*Case Mix Problem (CMP)*”, which involves the allocation of theatres to different surgical specialities over an extended time horizon. The tactical level involves medium-term planning, such as allocating theatre time slots to specialities and consultants/surgeons within a scheduling window. Known as the “*Master Surgery Scheduling (MSS)*” problem, this level aims to maximise and balance resource utilisation while addressing the overall requirements of specialities. Finally, the operational level deals with short-term decisions, such as assigning specific surgical cases to theatres and planning associated resources. This level, referred to as the “*Surgical Process Scheduling (SPS)*” problem, produces a detailed timeline for surgical cases, including start dates, times, locations, and the allocation of resources such as wards.

Again, scheduling approaches for tactical and operational planning can be broadly categorised into three types: block scheduling, modified block scheduling, and open scheduling [5]. In block scheduling, predefined time blocks are assigned to surgeons or consultants, and the corresponding resources are blocked in advance. This approach typically involves first addressing the MSS problem by assigning time blocks to surgeons, followed by filling these blocks with specific surgical cases at the SPS level. In contrast, open scheduling is a more flexible approach in which there are no predefined session-to-surgeon assignments, and the MSS and SPS problems are solved together [4, 6]. Modified block scheduling combines the rigidity of block scheduling with the flexibility of open scheduling, providing a hybrid approach that is both adaptable and convenient for planners.

Current planning methods face two major challenges related to the theatres’ utilisation and patient care: (1) surgery cancellations caused by time overruns in preceding procedures, and (2) under-utilisation of theatres when surgeries are completed in less time than allocated [7]. To alleviate these inefficiencies and reduce the average waiting time for patients on RTT lists, optimal planning of surgical theatres and resources is imperative [8].

Optimal surgical scheduling, on the other hand, is an inherently complex process that requires a delicate balance between business objectives and patient care priorities. It involves managing resource constraints and coordinating key participants, such as surgeons, nurses, and patients. Achieving these goals is challenging, as it must address the competing needs of hospitals, staff, and patients while enhancing the overall performance of surgical theatres.

Research studies are being pursued for digital advancement within the field of surgical scheduling, exploring and adopting methods such as simulation, analytics (including queueing theory), and mathematical optimisation [9]. However, real-world adoption remains limited because elective scheduling is dominated by uncertainty (particularly variable procedure durations and cancellations) and by unavoidable multi-objective trade-offs. In practice, stakeholders must simultaneously balance waiting-time reduction, Referral-to-Treatment service targets, fairness by clinical priority, theatre utilisation, staff overtime, and cancellation risk—objectives that can conflict under the same capacity constraints and can shift as conditions change. This combination of uncertain inputs and competing objectives is a key reason why many promising operational research results fail to translate into reliable, sustained decision support for elective theatre scheduling [10].

This study presents a novel integration of machine learning, statistical modelling, mathematical optimisation, and meta-heuristic algorithms to address the highly complex and resource-constrained problem of elective operating room scheduling, grounded in scenario-specific requirements and empirical data from an NHS Trust. By providing solution(s) applicable to a real healthcare setting, it contributes to improved efficiency and equity—supporting fair, timely reductions in Referral-to-Treatment (RTT) time and, in turn, improved patient satisfaction. Crucially, the proposed framework is designed to be implementable in routine NHS operations by producing transparent, constraint-feasible schedules that explicitly manage uncertainty and multi-objective trade-offs. Furthermore, this work lays the groundwork for a Healthcare Digital Twin framework that integrates machine learning, simulation, and optimisation for real-time decision-making.

The rest of the paper is structured as follows: Section 2 discusses various approaches and trends in surgical scheduling, focusing on the application of advanced data science, statistics, and applied mathematics. Section 3 outlines the methodology steps this research work follows for optimisation-based scheduling, including problem formulation and algorithm selection. Section 4 presents the experimental setup, their running and insights from the results. Section 5 concludes the paper and outlines future research directions.

2 Background and Related Work

The elective surgery scheduling problem has been a topic of interest in operations research for decades [4, 11, 12]. While decisions across strategic, tactical, and operational levels (Section 1) are all essential, the literature most commonly concentrates on operational decision-making—often framed as the Surgical Process Scheduling (SPS) problem—because it directly shapes day-to-day capacity use and patient flow [5, 13]. A smaller body of work addresses integrated planning and scheduling at the tactical level [14, 15]. This emphasis reflects the immediate consequences that tactical and operational decisions have for allocating scarce resources. In terms of how theatre time is organised, many hospitals and research studies adopt a block scheduling paradigm, largely because fully open scheduling introduces greater degrees of freedom and, consequently, higher modelling and implementation complexity [13]. Recently, modified

block scheduling has gained traction as it offers a pragmatic middle ground between flexibility and operational control [5].

During surgical scheduling at the operational level, it is desirable to ensure effective use of available day slots, smooth bed utilisation on the ward, allocation of surgeons to preferred sessions, and timely reduction of patient waiting time. Accordingly, the literature reports a wide range of performance measures, including patients' waiting times; utilisation, underutilisation, and overtime of operating theatres, wards, and intensive care units; throughput; patient deferrals; makespan (the finish time of the last surgery on the schedule); financial measures of resource use; and alignment with surgeons' preferences, among others [10, 16]. Depending on the metric, the scheduling task aims to minimise (e.g., waiting time, overtime, makespan) or maximise (e.g., utilisation, throughput) these objectives. Although multiple objectives are common, some studies model them as genuinely multi-objective problems [17], whereas others combine them using a weighted-sum formulation [18, 19]. Because institutional priorities differ across hospitals, a weighted formulation is particularly valuable as it allows decision-makers to reflect local policy and operational preferences by explicitly tuning the relative importance of competing goals [18, 19].

In the context of solution techniques for operating room planning and scheduling, they can be categorised into: analytical procedures (e.g., queueing theory), mathematical programming, dedicated branch-and-bound, scenario analysis (or sensitivity analysis), simulation and various heuristics [9]. Among these, focus of most researchers while developing solutions are mostly into: optimisation and/or simulation [20]. Following the concept of mathematical programming, many researchers have worked on integer/mixed-integer programming (MIP) models [21–23]. [21] offered a robust linear-programming approach to solve the mathematical optimisation for the surgical case assignment problem (SCAP), i.e., to determine the subset of patients to be scheduled in the considered time horizon and their assignment to the available OR block times. The aim was to minimise the penalty associated with the tardiness of the patients and with the waiting time. [23] aimed to find the optimal sequence and schedule of patients by minimising the cost of overtime, makespan and surgery completion time. Although mathematical programming is followed in this context by most researchers [9, 10], some also explored meta-heuristics approach, such as the implementation of genetic algorithm [12, 17] and simulated annealing [24].

While, more than half of the problems are formulated and studied as combinatorial optimisation problems, other approaches can also be found [10, 16]. Simulation models are preferred in the situation where multiple-stages: pre-surgical, surgical and post-surgical are dealt with, and/or each timing is represented as probabilistic distributions [9, 25]. However, running simulations for millions of combinations becomes computationally expensive; thus the focus is moreover towards mathematical programming. Moving a step towards advancement, the hybrid approach which leverages both simulation and mathematical programming or heuristic/algorithmic is expected to provide better solutions [9, 10].

One of the main scopes associated with scheduling strategies is the inherent uncertainty during surgery. Variables such as the duration of pre-surgery, surgery, and

patient recovery times are inherently uncertain and must be estimated [9]. Additionally, patients’ arrival time is uncertain to some extent. These uncertainties are also foreseen as a core reason for re-scheduling requirements, thus demanding more complex assessments necessary during scheduling. For example, [21, 26] treated the duration for surgery as an uncertain parameter. In contrast, some researchers assume fixed surgical durations for specific procedures [9, 27]. Sample average approximations are often preferred due to computational complexity in the stochastic assessment of the uncertain variable. In addition, alternative efforts have also been made to assess procedural time uncertainty in different ways. For instance, [22] and [28] incorporated overtime cost as one of the objectives, whereas [15] directly used the probability of overtime, calculated using historical data. However, these are mostly limited to the specialty level, i.e., considering all procedures together to have time-distribution under a particular specialty. The other aspects of data science, namely machine learning (ML) and deep learning, holds significant potential to address time-related uncertainties in surgical scheduling by enabling situation-based time estimation. However, the adoption of these techniques remains limited, and discussions of their application are still in their infancy [29, 30]. This limited adoption can be attributed to several factors, including a lack of confidence in ML’s ability to accurately predict time, the inherent complexity of surgical scheduling itself, and the added challenge of integrating these two distinct aspects.

Unlike many contemporary studies that examine predictive modelling, simulation, or optimisation in isolation, this work advances elective operating room scheduling through a hybrid decision-support pipeline that combines data-driven prediction with optimisation-based scheduling. This mixed-method design is specifically intended to address the practical complexity of the scheduling problem, where uncertain inputs and competing objectives must be handled within tight operational constraints.

The distinct contributions are as follows:

1. **ML-informed optimisation with uncertainty signals:** This study incorporates previously published and validated machine learning models for surgical duration prediction to strengthen the optimisation framework with procedure-level, data-driven estimates. In addition to point predictions, the trained models provide operationally relevant risk signals—such as the likelihood of session overruns or cancellations—derived from historical performance. These predictive components enable rapid, repeatable estimation of uncertainty-related quantities that would typically require time-intensive simulation–optimisation loops, thereby reducing turnaround time from hours to minutes in routine planning contexts.
2. **Two complementary optimisation strategies for different scales:** To ensure applicability across varying problem sizes, this study contributes a two-track optimisation framework and benchmarks both approaches across the full range of test instances. We employ the exact Integer Linear Programming solver SCIP (via PySCIPOpt) to generate high-quality, constraint-feasible schedules and demonstrate effectiveness. In parallel, we develop and evaluate a bespoke Simulated Annealing (SA) metaheuristic that provides a scalable and flexible alternative for larger or more computationally demanding instances, enabling efficient exploration of the solution space while maintaining practical feasibility. Unlike the

comparatively black-box solver behaviour of SCIP, SA also supports more transparent search dynamics (e.g., tracking incumbent improvements and neighbourhood moves), which helps practitioners inspect, diagnose, and refine scheduling trade-offs during experimentation and deployment.

3 Methodology

Considering the scope of scheduling at the operational level, the block scheduling strategy is adopted with the number and duration of available operating room blocks—referred to as sessions (Figure 1)—are predefined. Each session is associated with an allocated clinical team and represents a hard capacity limit for elective work on that list. The core scheduling decision is therefore formulated as an assignment problem: selecting and allocating patients from the waiting list to these predefined sessions so as to satisfy constraints and optimise the chosen objective(s).

We adopt a block scheduling strategy for three practical reasons:

- **Fit with operational practice:** A session-based template is already the standard unit of operational planning in many NHS elective settings; modelling decisions at the “session” level aligns the optimisation outputs with how lists are created, reviewed, and actioned, improving feasibility and acceptability.
- **Natural capacity-constrained formulation:** With fixed session capacity and variable case durations, the assignment resembles a knapsack-type selection problem, which enables a clear and computationally tractable mapping to exact and heuristic optimisation methods while retaining operational constraints.
- **Modularity for deployment and evaluation:** Working session-by-session supports scalable implementation, allowing incremental testing (by specialty, theatre, or time horizon) and easier auditing of objective trade-offs and constraint adherence in a live healthcare environment.

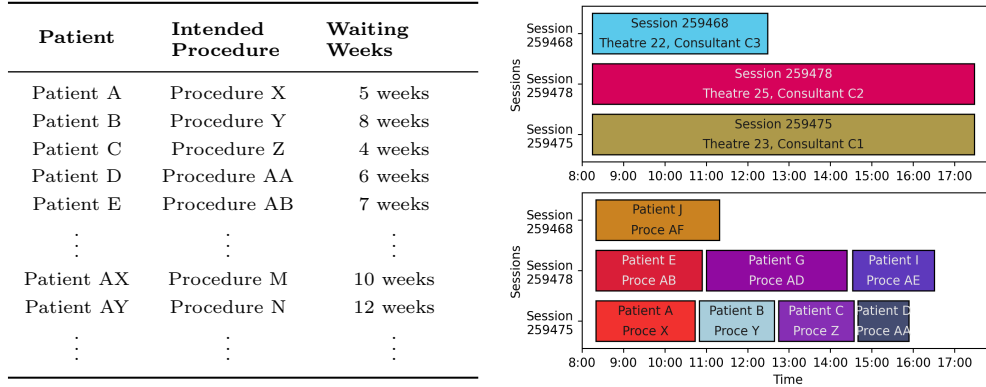


Fig. 1 Patient pool from waiting list (left), sessions' time blocks reserved for theatres & consultants including their team (upper-right), and consultants subjective block-schedule (patients booking) sample (lower-right) for the given sessions.

The block scheduling strategy assumes a fixed number of pre-allocated sessions for the clinical team, and the elective patients' related fixed waiting list throughout the scheduling. In addition, it is assumed that all operating rooms are identical in terms of resources and equipment, allowing any patient to be scheduled in any available room. The sequencing phase—i.e., the order in which patients undergo surgery within a session—is not considered, as the overall performance of the schedule is evaluated at the session level, making the internal arrangement of patients within a block irrelevant.

3.1 Problem Formulation

The optimisation problem formulation includes defining decision variables, objectives, and the constraints mathematically. The formulations aligning to the block-scheduling strategy are as follows:

3.1.1 Decision Variables

Let $P = \{p_1, p_2, \dots, p_n\}$ be the set of patients, where each $p_i \in P$ for $i = 1, \dots, n$, and let $S = \{s_1, s_2, \dots, s_m\}$ be the set of available session blocks, where each $s_j \in S$ for $j = 1, \dots, m$.

Solving the problem using a block scheduling strategy means determining whether a patient p_i is scheduled for a session block s_j . For this, a binary decision variable $x_{ij} \in \{0, 1\}$ is defined, where:

- $x_{ij} = 1$ if patient p_i is assigned to session s_j ,
- $x_{ij} = 0$ otherwise.

3.1.2 Objective Functions

At this stage, the elective surgery scheduling problem is formulated with three objectives that reflect both operational priorities and patient-centred outcomes. Specifically, we aim to (i) maximise utilisation of available theatre sessions' time, (ii) minimise the risk of session overruns (equivalently, maximise the likelihood that sessions finish within their allocated duration), and (iii) prioritise patients in a way that achieves the greatest reduction in waiting time. These objectives are aligned with commonly reported performance measures in the operating room scheduling literature and are consistent with practical hospital decision-making, supporting their relevance for real-world implementation [10, 16].

Objective 1—Theatre Utilisation (f_1)

The first objective is to maximise the utilisation of the available operating theatre time within each session. Intuitively, this encourages schedules that make effective use of the fixed session capacity by selecting an appropriate set of cases (with varying expected durations) so that unused time is reduced, while still respecting feasibility constraints. This objective is represented mathematically as:

$$f_1(x) = \sum_{j=1}^m \frac{1}{T_j} \sum_{i=1}^n t_{ij} \cdot x_{ij}$$

where,

- n : total number of patients,
- m : total number of session blocks,
- $x_{ij} \in \{0, 1\}$: binary decision variable; $x_{ij} = 1$ if patient p_i is assigned to session s_j , 0 otherwise,
- t_{ij} : time (or cost) associated with assigning patient p_i to session s_j ,
- T_j : total available time (or capacity) in session s_j ,

Objective 2— Maximising Session no Overrun Probability (f_2)

Minimising the likelihood of session overrun can be equivalently expressed as maximising the probability that the session finishes within its allocated time. In this study, this probability is quantified using the surgical-time estimates produced by pre-trained ML models together with their empirically observed prediction error on historical cases, thereby incorporating the model’s real-world uncertainty into the scheduling decision. Rather than relying solely on point estimates of duration, this additional objective explicitly accounts for the risk that accumulated case-time exceeds the session capacity, which is a key driver of overtime, disruption to downstream resources, and list instability.

Conceptually, this risk-aware formulation serves a similar purpose to confidence-interval-based scheduling, but it is grounded in observed predictive performance and is directly actionable within the optimisation framework. Detailed steps for computing the probability of no session overrun are provided in [Appendix A](#).

Objective 3—Reducing RTT Waiting Times (f_3)

Reducing Referral-to-Treatment (RTT) waiting time is a central objective for elective services, as prolonged waits can adversely affect patient experience and, potentially, clinical outcomes. In this study, f_3 is formulated to ensure that the scheduling decisions actively contribute to backlog reduction in an equitable manner by preferentially selecting patients who have waited longer. This prioritisation is operationalised by assigning higher preference to higher-wait cases, so that, all else equal, long-wait patients are more likely to be scheduled earlier.

At the same time, f_3 incorporates a normalisation of waiting times to avoid disproportionate influence from extreme values and to maintain fairness across the overall case-mix (e.g., across procedure types where baseline waiting-time distributions differ). This normalisation also supports the multi-objective setting by placing the waiting-time objective on a commensurate scale with the other objectives prior to weighting, so that the chosen weights reflect intended policy trade-offs rather than artefacts of measurement units or magnitude.

$$f_3(x) = \frac{1}{\sum_{i=1}^n WT_i} \sum_{i=1}^n WT_i \left(\sum_{j=1}^m x_{ij} \right),$$

where:

- n : total number of patients,

- m : total number of session blocks,
- WT_i is the waiting time of patient p_i ,
- $x_{ij} \in \{0, 1\}$ is a binary decision variable indicating whether patient p_i is assigned to session s_j ,

Note: $f_3(x)$ would be computed using waiting times (WT) obtained at the scheduling time-stamp, rather than making WT session-specific. This simplification does not affect the optimisation problem, as proportional changes in WT impact both the numerator and denominator equally in this ratio-based objective.

3.1.3 Multiple Objectives Assessment

The optimisation framework considers three objectives simultaneously; however, the way trade-offs are handled depends on the solution technique.

Pareto-based multi-objective formulation

When objectives are treated explicitly as a vector, the problem is posed as a multi-objective optimisation:

$$\max_x (f_1(x), f_2(x), f_3(x)), \quad (1)$$

where solutions are compared using Pareto dominance. In this setting, no single solution is universally optimal; instead, the outcome is a set (or archive) of non-dominated solutions that represent different efficiency–risk–equity trade-offs. For example, aggressively prioritising long-wait patients may reduce utilisation, whereas pushing utilisation higher may increase overrun risk.

This formulation supports transparent trade-off exploration by presenting multiple candidate schedules for decision-maker selection. However, maintaining and exploring a non-dominated set can become computationally expensive as problem size increases.

Weighted-sum scalarisation

When an explicit Pareto-based search is impractical or when using solvers requiring a scalar objective (e.g., SCIP), the objectives are combined into a single weighted function:

$$\max_x w_1 f_1(x) + w_2 f_2(x) + w_3 f_3(x), \quad (2)$$

where w_1, w_2, w_3 reflect the relative importance of each objective. Prior to weighting, objectives are normalised to ensure that the weights represent preference trade-offs rather than differences in scale across metrics.

This approach yields a single implementable schedule and is computationally efficient, but it can obscure trade-offs and is sensitive to the choice of weights; therefore, weight selection should be guided by stakeholder priorities and accompanied by sensitivity analysis where possible.

Hybrid Approach: Penalised Utilisation with Independent Secondary Objective

In this problem setting, **session utilisation** (f_1) and **probability of no overrun** (f_2) exhibit an inverse relationship—higher utilisation tends to increase overrun probability (f_2), with $\bar{f}_2 = 1 - f_2$. In other words, the inversely correlated objectives

(f_1 and f_2) are conditionally dependent: overrun risk becomes relevant only when a session is meaningfully utilised, and when utilisation is low, overruns are unlikely.

To reflect this feature, we adopt a hybrid multi-objective formulation, where Objectives 1 and 2 are integrated into a single penalised scalar objective (F_1), and Objective 3 is retained as an independent objective in a two-objective optimisation framework. The combined objective is expressed as:

1. **The penalty formulation:**

$$F_1 = f_1 - \lambda \bar{f}_2 = f_1 - \lambda(1 - f_2) \quad (3)$$

Since the constant term $-\lambda$ does not affect optimisation, we obtain the equivalent:

$$F_1 = f_1 + \lambda f_2 \quad (4)$$

2. **Weight Relationship:** The penalty factor λ can thus be interpreted as the ratio of weights:

$$\lambda = \frac{w_2}{w_1} \quad (5)$$

3. **Hybrid Objective Representation:** The final hybrid problem, derived from Equations 1 and 2, becomes:

$$\max_x \left(w_1 f_1(x) + w_2 f_2(x), \quad f_3(x) \right) \quad (6)$$

where w_1, w_2 are weights for the aggregated objectives, and f_3 is treated separately.

Also the Equation 6, can be represented as:

$$\max_x \left(F_1(x), f_3(x) \right) \quad (7)$$

By integrating session utilisation and overrun risk into a single penalised objective, the model eliminates the distortion caused by treating inversely related objectives independently, thereby avoiding convergence toward trivial, low-utilisation solutions. This scalarisation simplifies the optimisation landscape while preserving the operational relevance of both efficiency and safety.

3.1.4 Constraints— Solution Feasibility Conditions

The optimisation process must adhere to several constraints while generating solution(s) ensuring their feasibility and reflecting operational requirements to real-world resource limitations.

Variable type Constraints: The decision variable is always binary represented by 1 and 0, for whatever size it is.

$$x_{ij} \in \{0, 1\}, \quad \forall i = 1, \dots, n, \quad \forall j = 1, \dots, m \quad (8)$$

Patient Assignment Constraint: Each patient is assigned to at most one surgical session, as enforced by:

$$\sum_{j=1}^m x_{ij} \leq 1, \quad \forall i = 1, \dots, n \quad (9)$$

Session Capacity Constraints: No session exceeds its allocated time while accounting for:

- Maximum theatre utilisation ($uti_{\max}$)
- Procedure-specific turnover time (γ_{ij})

$$\sum_{i=1}^n t_{ij} \cdot x_{ij} \leq uti_{\max} \cdot T_j, \quad \forall j = 1, \dots, m \quad (10)$$

$$\sum_{i=1}^n (t_{ij} + \gamma_{ij}) \cdot x_{ij} \leq T_j, \quad \forall j = 1, \dots, m \quad (11)$$

Operational Considerations:

- **Turnover Time:** The inclusion of γ_{ij} reflects real-world requirements for preparation and cleaning between procedures, which are not captured in utilisation KPIs (f_1) but directly impact scheduling feasibility.
- **Utilisation Management:** The parameter $uti_{\max} < 1$ ensures buffer time for clinical operations, acknowledging that 100% theoretical utilisation is neither practical nor desirable in surgical settings.

3.2 Machine Learning for Time Estimation

Higher accuracy on surgery time predictions is anticipated to minimise inefficiencies such as session overruns or underutilisation, which are commonly caused by subjective estimates or reliance on historical averages. For this, pre-trained machine learning (ML) models are employed.

The ML models used in this study were developed and validated as part of previously published work in [31]. That work implemented and compared multiple ML algorithms, including Neural Networks, ElasticNet regression, Gradient Boosting, and Bayesian Ridge regression, for surgery time estimation in Trauma and Orthopaedics procedures within an NHS Trust hospital. Specifically, the models were trained on four years of elective Trauma and Orthopaedics surgery data from the NHS Trust hospital, using only patient- and hospital-setting features that are available at the time of theatre planning, to improve the practical realism of the predictions. Though the findings have already demonstrated robust predictive performance, the performance is again validated with new dataset.

This integration of ML ensures that the scheduling optimisation process benefits from data-driven predictions, enabling more accurate and efficient resource allocation.

3.3 Optimisation Solution Methods

Two distinct optimisation methodologies were implemented: one based on Integer Linear Programming (ILP) and the other on an algorithmic metaheuristic approach.

The purpose of using two different methodologies is to exploit the strengths of each method, and conduct a comprehensive comparative analysis to understand their performance across various aspects of the problem. This comparison would provide valuable insights into the suitability of each method for different problem scales, objectives, and practical constraints.

3.3.1 SCIP Solver

To capitalise on the benefits of ILP, which are particularly effective for problems requiring mathematical rigour and exact solutions, the constraint integer programming solver SCIP [32], was adopted. The reason for selecting SCIP being that it is one of the advanced solvers for linear and combinatorial optimisation problems, making it well-suited for surgical scheduling where constraints such as theatre capacity, surgeon availability, and session duration must be strictly respected. Its advanced features include: branch-and-bound, cutting planes, and primal heuristics, which are beneficial to address the complex combinatorial nature of the scheduling problem. The SCIP solver systematically explores the solution space, guaranteeing globally optimal solutions for small to medium problem sizes. However, as the problem size grows, the computational effort required by ILP solvers increases significantly, potentially making this approach impractical for larger datasets.

To operationalise the SCIP-based ILP formulation within a reproducible Python workflow, we adopted PySCIPOpt [33], which provides a Python interface to SCIP and enables integration with other Python-based subsidiary tools. In this setup, SCIP serves as the solver backend: the internal search is managed by the optimiser, while we retain full control over the mathematical formulation, data inputs, and solution post-processing in Python. Practically, we additionally enforce explicit time limits to reflect operational decision-making constraints and to ensure consistent runtime across instances. Necessary adaptations are implemented at the modelling layer; in particular, the multi-objective formulation is scalarised into a single-objective ILP via weighted coefficients (Appendix A) to ensure compatibility with the ILP framework and enable explicit trade-off tuning.

3.3.2 Simulated Annealing (SA)

Simulated Annealing (SA) represents a meta-heuristic algorithmic approach designed to explore complex solution spaces efficiently. Unlike SCIP, which is limited by linearisation and computational intensity for large problems, SA is inherently flexible and scalable. It does not require the problem to be transformed into a linear form, making it well-suited for addressing multi-objective optimisation directly. Furthermore, this algorithm’s concept is to balance exploration and exploitation by probabilistically accepting worse solutions early in the search process to escape local optima. This flexibility enables SA to provide high-quality near-optimal solutions for larger problem instances where exact solvers struggle.

Each run of the SA-based Algorithm 1 yields a Pareto archive, capturing a diverse set of trade-off solutions across the objectives. These archives represent non-dominated solutions, offering decision-makers a spectrum of choices balancing criteria without a single objective being strictly optimal.

Algorithm 1 Simulated Annealing for Multi-Objective Surgical Scheduling

Require: Initial temperature T_0 , cooling rate α , stopping criteria ($T_{\min}$ or $n_{\max}$)

- 1: Initialise current solution x_{current} with an empty schedule or any feasible schedule
- 2: Evaluate objectives F_1, f_3 for x_{current}
- 3: Set $T \leftarrow T_0, n_{\text{iter}} \leftarrow 0$
- 4: Set **Pareto_archive** $\leftarrow \emptyset$,
- 5: Set **Improvement_log** $\leftarrow \emptyset$
- 6: **while** $T > T_{\min}$ **and** $n_{\text{iter}} < n_{\max}$ **do**
- 7: **Generate a new solution** x_{new} :
- 8: Randomly choose one or more neighbourhood operations:
- 9: (a) add a patient to a session
- 10: (b) add a patient and delete another from the session
- 11: (c) Move a patient between two sessions (one-way)
- 12: (d) Swap two patients between two sessions (two-way)
- 13: Apply the selected operations to x_{current} to create x_{new}
- 14: Evaluate objectives F_1, f_3 for x_{new}
- 15: **if** x_{new} Pareto-dominates x_{current} **then**
- 16: Set acceptance probabilities- $P_1 \leftarrow 1, P_2 \leftarrow 1$ for F_1 and f_3 respectively:
- 17: **else**
- 18: Compute acceptance probabilities- P_1, P_2 for F_1 and f_3 respectively:

$$P = \exp\left(\frac{f(x_{\text{new}}) - f(x_{\text{current}})}{T}\right)$$

- 19: **end if**
 - 20: Generate random numbers, $r_i \in [0, 1], \quad i \in \{1, 2\}$
 - 21: **if** $\text{any}(r_i < P_i)$ **then**
 - 22: $x_{\text{current}} \leftarrow x_{\text{new}}$
 - 23: Update **Pareto_archive** with x_{current}
 - 24: Add x_{current} to the **Improvement_log**
 - 25: **end if**
 - 26: Update temperature: $T \leftarrow T \cdot \alpha$
 - 27: Increment iteration counter: $n_{\text{iter}} \leftarrow n_{\text{iter}} + 1$
 - 28: **end while**
-

We developed a custom Python implementation of SA (Algorithm 1), allowing transparent tracking of the search trajectory and straightforward adaptation of operators and parameters to NHS-style session-based constraints typical of NHS elective theatre scheduling. The resulting SA solver is tailored to the structural requirements of the surgical scheduling problem and supports the hybrid objective formulation in Equation 7, by maintaining an archive of non-dominated solutions. Its flexibility in neighbourhood design and parameter tuning makes it well suited to larger instances or tighter runtime budgets, where exact optimisation can become computationally prohibitive.

4 Experiments and Results

This study focuses on elective patient scheduling at the *Essex and Suffolk Elective Orthopaedic Centre (ESEOC)*, restricted to the *Trauma and Orthopaedics (T&O)* specialty. The dataset comprises elective waiting-list records together with pre-allocated future operating sessions at ESEOC, providing a realistic and representative operational scheduling setting. In addition, historical activity data are used to estimate procedure durations and quantify overrun risk (Section 3.2).

4.1 Model Validation on Time Estimation

The scheduling optimisation framework relying on machine learning (ML)-based predictions of surgical durations called for a rigorous evaluation of the accuracy of these predictions. Therefore, a discrete experiment was conducted to validate the ML models’ performance by comparing the **model-predicted surgery times** with a scheduling optimisation framework relying on **machine learning (ML)-based predictions** of surgical durations called for a rigorous evaluation of the accuracy of these predictions. Therefore, a discrete experiment was conducted to validate the ML models’ performance by comparing the **model-predicted surgery times** with the **actual surgical durations**, recorded as “H4 Minutes”.

4.1.1 Experimental Setup

This analysis focuses on Trauma and Orthopaedics (T&O) surgical procedures, both planned and performed at ESEOC over a three-month period. Schedules were generated monthly, each covering a one-month planning horizon, using the model’s estimated durations (averaged when multiple models were employed). All consultant–patient allocations adhered strictly to the consultants’ pre-established schedules.

Once actual surgical data became available, a comparative analysis was performed between the consultant-planned durations, the model-predicted durations, and the actual recorded durations. Only those cases present across all three datasets—the consultant’s plan, the model-generated schedule, and the actual surgical records—were included in this evaluation. A total of 1,212 surgical cases met these inclusion criteria.

4.1.2 Analysis and Findings

Cases are sorted by real surgical duration, offering a clear perspective on discrepancies and alignment across the dataset. The scatter plot in Figure 2 visually compares the actual surgical durations (H4 Minutes) with the durations planned by consultants and those estimated by the model.

A clear spread is observed between the actual durations (H4 Minutes) and both the consultant planned times and the model estimates, highlighting substantial variability in procedure timing. Consultant plans appear more discrete, reflecting reliance on a smaller set of inputs (typically procedure type/code and clinical judgement), whereas the ML model incorporates a broader set of predictive features and, therefore, produces a wider range of estimates. From a planning perspective, consultant plans tend

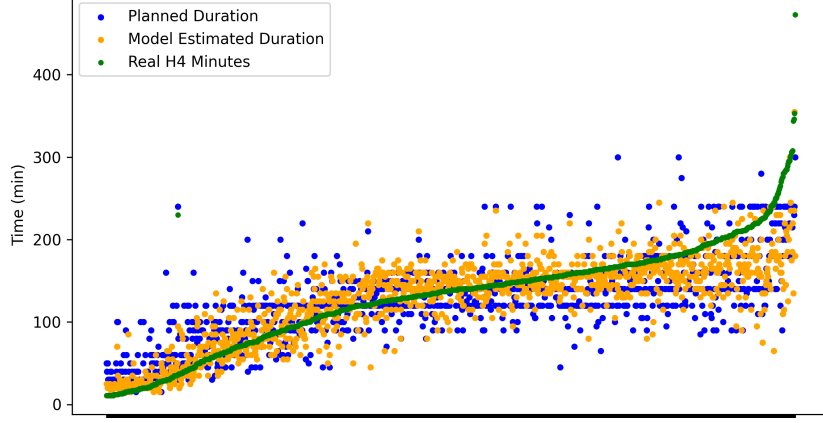


Fig. 2 Comparison of Surgical Durations: Real vs. Consultant Plan vs. Model Estimation

to overestimate shorter procedures and underestimate longer ones, consistent with a regression-to-the-mean effect; in contrast, the model exhibits a more heterogeneous pattern with both under- and overestimation across the duration range. For shorter procedures, the model's predictions are more tightly clustered around the actual durations, suggesting improved alignment and reflecting its strength in estimating less complex cases. While the model effectively captures general trends in surgical timing, some variability persists, likely due to unaccounted clinical factors.

This difference is quantitatively supported by error metrics. The Root Mean Square Error (RMSE) for the consultant plans is 40.42 minutes, compared to 35.3 minutes for the model estimates. Additionally, the Mean Absolute Percentage Error (MAPE) is significantly lower for the model at 28.06%, compared to 47.77% for the consultant plans. These results indicate that the model provides a more accurate and consistent estimation of surgical durations.

Overall, there remains scope for further refinement of the model to reduce prediction errors. Nevertheless, given the model's demonstrated proximity to real-world values and its superior performance over consultant plans, the trained models were used to estimate surgical durations prior to solving the optimisation problem.

4.2 Optimisation Experimental Setup

The optimisation based surgical scheduling problem was designed with few considerations and assumptions. To thoroughly evaluate the computational performance of the two optimisation approaches, multiple problem sizes were selected:

- **Size A:** 54 surgeries and 12 sessions (size of x : 648).
- **Size B:** 100 surgeries and 20 sessions (size of x : 2000).
- **Size C:** 186 surgeries and 38 sessions (size of x : 7068).
- **Size D:** 302 surgeries and 59 sessions (size of x : 17818).

By using multiple problem sizes, this study offers valuable insights into the performance trade-offs between precision and scalability across varying problem complexities, which is essential for real-world applications where problem size can fluctuate significantly. Additionally, a **maximum session utilisation constraint** of 90.0% was enforced initially to accommodate uncertainties as preparation time.

For the SCIP solver, the scalar objective function was defined as $0.7 * f_1 + 0.2 * approx.(f_2) + 0.1 * f_3$, equivalent to the objective 2. Detailed on $approx.(f_2)$ can be found in Appendix A.2. The weights were determined through sensitivity analysis, aiming to ensure that each objective meaningfully influences the final solution, while minimising the extent to which optimising one objective compromises the others. Likewise, the stopping condition for the SCIP solver was provided as the time limit.

SA, on the other hand, followed a hybrid approach (Equation 6), with w_1 and w_2 being 0.75 and 0.25 respectively. The initial solution for the Algorithm 1 was considered to be an empty schedule. Likewise, the stopping condition for simulated annealing was set based on the final temperature, with parameters set as follows: *initial temperature* = 100,000, *final temperature* = 0.00001, and *alpha* = 0.999. These hyper-parameters associated with the algorithms are chosen after experimentation and analysis, and remain the same for all the problem sizes.

Due to its scalar objective formulation and deterministic nature, SCIP yields a single, consistent solution across repeated runs, provided no hyper-parameters are modified. In contrast, the SA solver produces a diverse set of solutions in the form of Pareto archives, with variability observed across runs even under identical settings.

Considering the stochastic nature of SA, we performed multiple independent runs of the optimisation algorithm to evaluate its performance across multiple trials.

4.3 Optimisation Results and Discussion

This section presents the optimisation results and their analysis from multiple perspectives, including the impact of increasing problem sizes, the consistency of performance achieved by the stochastic meta-heuristic, and the model’s capacity for patient selection based on assigned weightings and modified objectives.

For these analyses, the system used for experimentation was an HP Elite equipped with a 13th Gen Intel® Core™ i7-1365U 1.80 GHz processor and 32.0 GB of installed RAM.

4.3.1 SA Based Optimisation:

Given the stochastic nature of the SA solver, SA-based optimisation performance was analysed against algorithmic capacity to converge to a solution and effect by problem size. These analyses were made with solution aggregation from multiple optimisation runs.

Convergence and Variability Analysis:

Whenever a candidate (new) solution leads to an update of the Pareto archive during an SA optimisation run (Algorithm 1), it indicates that it is non-dominated and

contributes to improving the current approximation of the Pareto front. The objective values for the candidate solution whenever the Pareto archive was updated were recorded alongside the corresponding iteration count, to observe the trend of solutions and convergence over iterations. This process yields a record of the iterations at which meaningful improvements to the Pareto front occur.

Then, to assess the consistency/variability performance of the SA algorithm in converging towards the solution, the evolution of the Pareto front was monitored across the multiple independent optimisation runs. For instance, plotting the recorded objective values (either as individual objectives or as scalarised combinations) against the iteration count for 20 independent optimisation runs, the convergence behaviour and variability of the algorithm can be visualised (Figure 3). This plot provides insight into the algorithm’s stability, convergence dynamics, and robustness under stochastic influences.

For this analysis, one representative problem size i.e., (100, 20) was considered. Data points for the analysis incorporate both the objective functions (F_1 and f_3) over **23,000+ iterations across 20 independent optimisation runs**, for the problem.

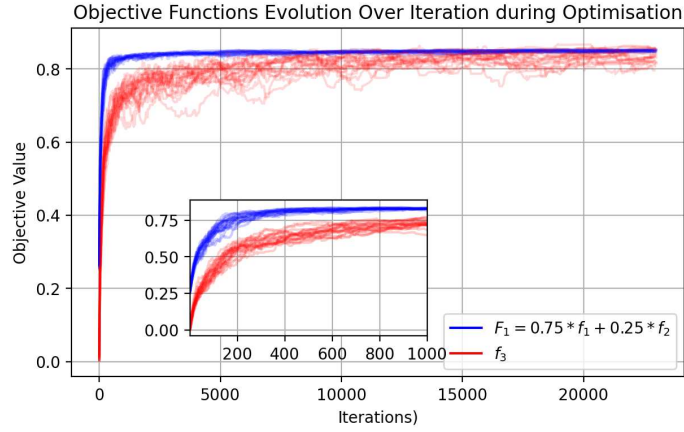


Fig. 3 Line plot of objective values of candidate solutions contributing to optimisation run progression (pareto-archive update) against 23,000+ iterations. Plot results to a fringe-like pattern when plotted for 20 independent optimisation runs.

Figure 3 shows distinct convergence behaviours forming a fringe-like pattern that captures both overall trends and variability. The solution trend shows that F_1 stabilised relatively early, indicating that the simulated annealing process quickly identifies favourable solutions within the neighbourhood search. In contrast, the f_3 gradually improves, reaching an intermediate plateau where progress stagnates before making further gains in the later iterations. This staggered behaviour suggests that the search initially struggles to escape certain solution neighbourhoods but eventually finds better alternatives as the temperature parameter decreases, allowing for more

refined exploration. As the overall spreads indicate the stochastic nature of the optimisation process, the narrow spread from the initial to the later iterations indicates consistency in performance.

Pareto Archive Analysis across different Problem Sizes:

To assess the performance across the problem sizes discussed earlier, Pareto archives were collected from 20 independent optimisation runs conducted under identical problem configurations and solver settings. Each run generated a distinct Pareto front, reflecting the inherent stochasticity of the meta-heuristic process. The solutions from all runs were aggregated to form a cumulative set, which was then visualised using hexbin density plots in the objective space (Figure 4). This approach enabled the identification of regions with high solution recurrence, indicating zones of convergence, as well as sparser regions, which reflect variability or less stable trade-offs.

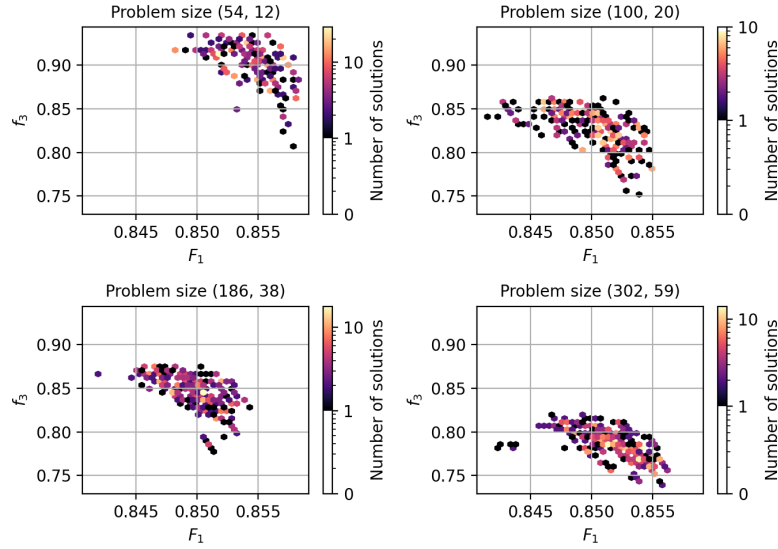


Fig. 4 Pareto solution distributions sub-plots with data from 20 Simulated Annealing runs; each sub-plot for different problem size. Each subplot shows a hexbin density map in the objective space, illustrating the spread and concentration of non-dominated solutions.

Further analysis across varying problem sizes revealed a distinct shift in the location of the Pareto front. As problem size increased, the set of non-dominated solutions moved consistently toward lower values across both objectives. This downward shift suggests that larger problem instances lead to comparatively lower values but less distributed Pareto solutions, potentially due to increased scheduling constraints or resource saturation. The results highlight how increasing complexity alters the structure of optimal solutions, requiring the solver to navigate a bigger solution space.

Solutions against the objectives:

In the above experiments, two objectives F_1 and f_3 were considered, with weightage for f_1 and f_2 taken as 0.75 and 0.25 respectively to have one objective (F_1). For deeper solutions' quality, all three objectives (also KPIs for schedules)— f_1 , f_2 and f_3 were computed separately for each individual in the pareto-archives, although optimisation runs considered the two objectives. The objectives were obtained from each of the 20 optimisation runs, and their distributions were separately analysed.

The box plot (Figure 5) demonstrates the KPIs distribution of each individual in pareto archives. The problem sizes which are represented on the x-axis correspond to an increasing number of patients and sessions, with the notation (n, m) indicating the number of patients and sessions, respectively. In addition, the average computation time for each problem size is displayed above each group.

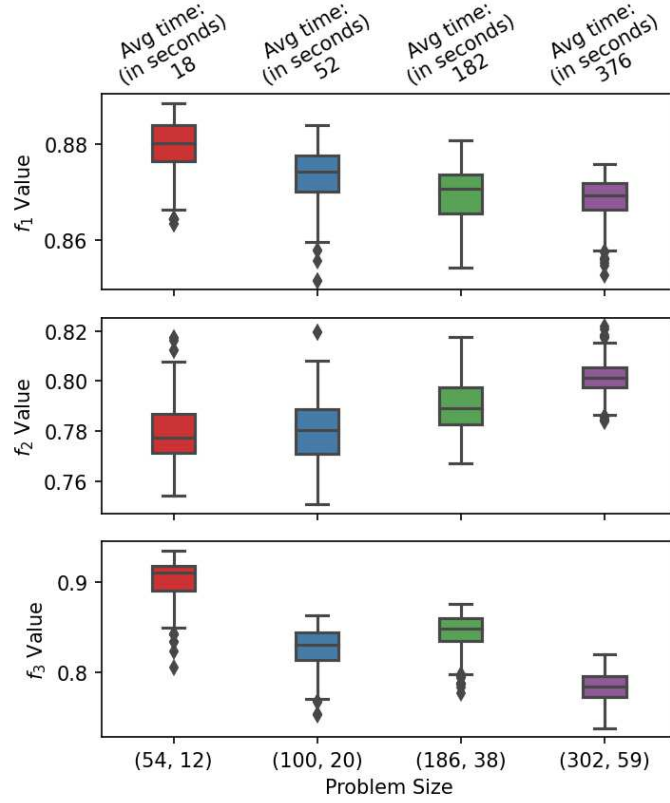


Fig. 5 Box plot of KPI distributions for Pareto-archive members across different problem sizes (cases count, sessions count). Results are based on an archive accumulated over 20 optimisation runs for each problem size. The average solution time (in seconds) is shown on the upper x-axis for each problem size.

Analysing the effect of increasing problem size on objective values for f_1 (solution quality in terms of utilisation), the values start relatively high for smaller problem sizes and tend to decrease slightly as the problem size increases. This suggests that maximising session utilisation becomes more challenging with a larger number of patients and sessions. The variability in the f_1 values remains moderate for sizes of the problem, indicating relatively consistent performance despite increasing complexity.

The observation of f_2 (probability of no overrun) values show an upward trend as the problem size increases. This somewhat justifies its relation with maximisation of session booking time or with f_1 , but in the opposite way.

For f_3 , the effect of increasing problem size has the similar trend as for f_1 trend. This suggests that maximising normalised waiting weeks becomes more challenging with a larger number of patients and sessions.

In terms of computational time and trade-offs, the average computation time increases significantly with problem size, from $\approx 15 - 20$ seconds for (54,12) size to $\approx 350 - 400$ seconds for (302,59) size. This reflects the growing complexity of solving larger optimisation problems, where scheduling constraints expand combinatorially. The trade-off between solution quality and computation time is evident: while larger problems take considerably longer to solve, the objective values show some degradation, particularly for f_1 and f_3 .

4.3.2 SCIP Based Optimisation:

The other optimisation approach considered was SCIP-based optimisation and the set up followed is already discussed above. Beside the hyper-parameters and problem size already discussed, the time limits for SCIP-based optimisation runs were set. One set of experiments was with the time matching the time taken for the corresponding problem sizes during SA-based optimisation runs, while other experiments allowed the algorithm to run for a maximum of one hour for the possibility of reaching the global minima.

The SCIP-based optimisation demonstrated consistent performance across problem sizes, achieving high-quality solutions (Figure 6). This consistency was maintained even when analysing each objective separately, despite the problem being solved using a scalar-weighted objective. Regarding the effect of increasing problem size on objective values, for f_1 , although the trend looks similar with the SA tending to decrease slightly as the problem size increases — the decrease is negligible.

4.3.3 Optimisation solution behaviour by solution approach

From a solution-method perspective, the ILP formulation solved with SCIP delivers consistently high-quality objective values across the tested instance sizes, with limited variability in outcomes under a fixed configuration (e.g., time limits). The bespoke SA metaheuristic, although able to generate feasible solutions across the same instances, typically attains slightly weaker best objective values and exhibits greater sensitivity to hyper-parameter settings and stochastic search effects.

In terms of solution structure, SCIP tends to produce solutions that more strongly reflect the intended trade-offs encoded by the scalarised objective, including clearer

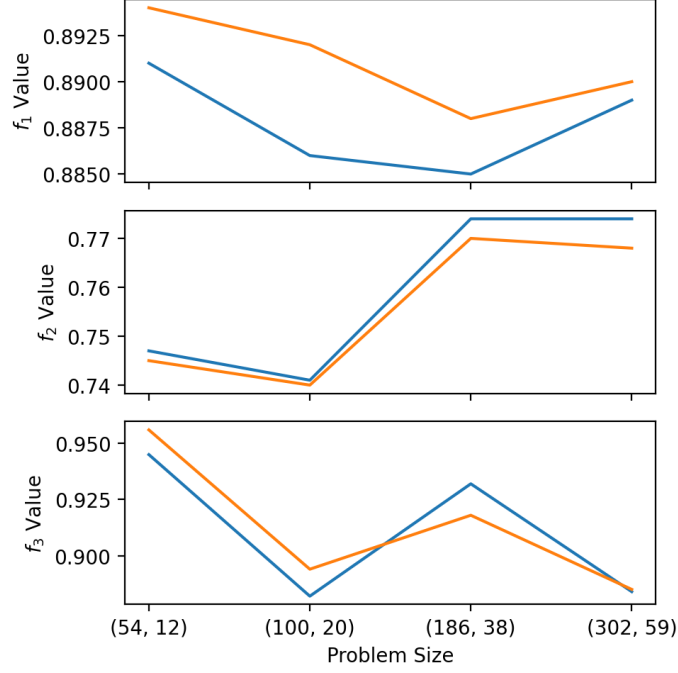


Fig. 6 SCIP performance with line plot for reached solutions’ KPIs for different problem sizes i) blue line for each experiment set with time-limit with corresponding SA related experimental time, ii) amber line when maximum time limit set to 1 hour.

RTT-driven prioritisation when weighted accordingly. SA provides competitive alternatives but with less sharp separation in objective attainment, particularly where multiple objectives compete closely.

At the same time, SA offers practical advantages for optimisation development and adaptation: its white-box implementation supports direct inspection of the search process, lightweight modification of neighbourhood operators and acceptance rules, and rapid reconfiguration as assumptions or constraints evolve. In addition, SA can incorporate objective components such as f_2 directly without requiring linearisation, whereas SCIP necessitates linear (or linearised) formulations for ILP compatibility.

Overall, SCIP performs best in terms of solution quality and consistency in our experiments, while SA provides a more flexible and extensible optimisation engine for scenarios where rapid adjustment, transparency, or non-linear objective handling is prioritised.

4.4 Schedule Performance Metrics and Prioritisation Assessment

Objectives 1 and 2 are directly interpretable from the schedule itself—how much session time is filled (utilisation) and how likely the list is to finish on time (no-overflow probability). Objective 3 is less immediately observable because “good” f_3 values do

not, on their own, reveal which patients were selected, nor whether the resulting prioritisation is equitable across the waiting list. We therefore complement the objective values with additional analyses that focus specifically on patient selection, examining whether the optimisation outcomes consistently prioritise higher-wait patients in a fair and transparent manner.

A stepwise scheduling procedure was implemented to evaluate fairness in patient selection across successive planning horizons. At Stage 1, the full set of waiting-list patients was included, but scheduling was performed against only an initial subset of sessions (i.e., the earliest portion of the overall time horizon). Once these sessions were filled, unscheduled patients were carried forward and scheduled at Stage 2 using the next tranche of newly available sessions; this process was repeated until all sessions under consideration had been allocated.

To isolate the effect of progressively expanding session capacity, we did not introduce new patients at later stages. Reintroducing additional cases at each stage would keep the case-to-session ratio broadly constant and would obscure the natural shift in selection pressure as capacity increases over time; therefore, only sessions were added stepwise, while the remaining patient pool was allowed to diminish across stages.

This stepwise setup enables granular inspection of model behaviour by revealing how the optimiser prioritises patients at each scheduling stage. It also induces natural variation in problem size—particularly for the SA-based approach—because its stochastic search may lead to different numbers of patients being scheduled at each stage, resulting in slightly different residual problem instances and a wider set of scenarios for analysis. Overall, the procedure reflects a realistic operational setting in which the scheduling horizon expands as additional sessions become available over time.

Under this sequential process, we expect the following patterns to emerge:

- Session utilisation should be highest in earlier stages, because the larger initial patient pool provides greater flexibility to assemble well-fitting case mixes.
- Patients with longer waiting times should be scheduled earlier in the process.
- The average waiting time of scheduled patients should decrease in later stages, as the longest-wait patients are progressively removed from the pool.

Observing these trends would indicate that the model supports urgency-based prioritisation and fairness in staged scheduling scenarios.

Although the primary motivation for the stepwise design is detailed behavioural analysis, it offers an additional practical benefit by decomposing the overall planning horizon into smaller sub-problems. This decomposition can reduce computation time and, in some settings, improve solution quality, which is relevant for both SCIP- and SA-based approaches when addressing larger instances over longer planning horizons or across multiple theatre lists.

For this study, a representative problem size of 200 patients and 40 sessions (200, 40) was chosen. The scheduling was executed in four steps, with 10 new sessions introduced at each stage to accommodate the patients who remained unscheduled in the previous stages.

In real-world scenarios, adding new sessions typically reflects the allocation of additional resources over an extended time horizon. As a result, patient waiting times would naturally increase when moving forward to these newly considered time-horizon. However, from the perspective of the third objective function (f_3), this change in waiting time does not alter the optimisation landscape. Although introducing new sessions in later stages increases the RTT waiting times for all patients, this increase affects both the numerator and denominator of the ratio-based objective f_3 (Section 3.1.2). Since the objective is being maximised, and the change is proportional across all patients, the relative behaviour of the function—and hence the optimisation outcome—remains unchanged. Therefore, RTT weeks were held constant in this experiment and waiting time (WT) is thus considered from the current time-stamp perspective, rather than being adjusted dynamically for different session availabilities.

4.4.1 Schedule(s) and KPIs from SCIP-based Optimisation

Given the deterministic nature of the SCIP solver, a single experimental run was conducted for the SCIP-based optimisation. The session utilisation for each of the 10 added sessions, their chances of overrunning, and the RTT waiting times of patients scheduled at each step were recorded. The distributions of these metrics were then analysed. Figures 7 and 8 illustrate the performance of the SCIP-based scheduling

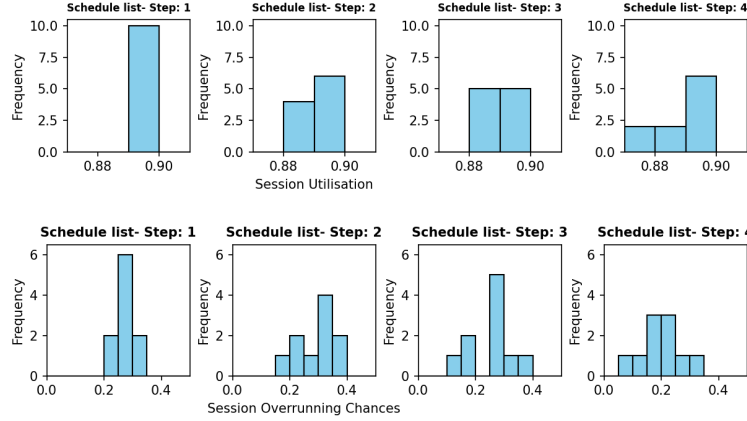


Fig. 7 Distribution of session-related KPIs for each session considered at different scheduling steps with a SCIP-based optimisation run. Top: session utilisation. Bottom: session overrunning probability.

method. The first one includes sessions related KPIs, while the second addresses KPIs related to patient prioritisation. Both figures include subplots showing the KPIs of the schedule generated at each step (Steps 1 to 4), excluding items already scheduled in previous steps. In the session-related KPIs, f_1 equals 0 and f_2 equals 1 when no scheduling has taken place. In contrast, patient waiting times can be measured both before and after scheduling. These are shown in Figure 8 under “Initial List” and “Unscheduled”.

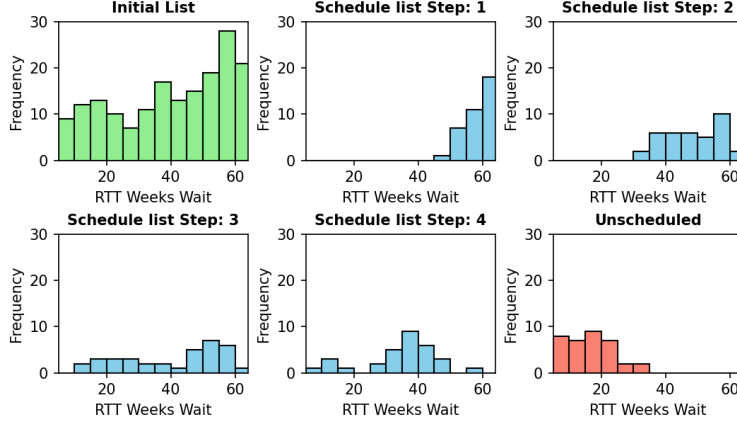


Fig. 8 RTT waiting time distribution (in weeks) of scheduled patients across different solution steps from a SCIP-based optimisation run. Distributions for the initial patient list and the unscheduled patients are also included.

Although the results come from a single, deterministic run, they clearly reflect the expected trends. In the first stage, session utilisation is relatively high. This is due to the availability of more scheduling options and variable case durations, despite the larger problem size (Figure 7, top). In Step 2, this benefit diminishes. Utilisation values become more distributed, but generally remain high as the problem size decreases in subsequent steps (Steps 2 to 4). Regarding the likelihood of session overruns, the trend appears to correlate with session utilisation, as shown in Figure 7(bottom).

Figure 8 demonstrates a clear prioritisation of patients with longer waiting times in earlier steps, with average waiting times decreasing in later steps. The unscheduled patients primarily consist of those with relatively shorter waits, indicating fairness-based scheduling.

4.4.2 Schedule(s) from SA-based Optimisation

In contrast to the deterministic nature of the SCIP solver, 10 independent experimental runs were conducted for the SA-based approach to account for its stochastic behaviour. The populations from each solution step were then pooled across runs (resulting in an expected dataset size 10 times larger than that of the SCIP-based approach) to enable trend analysis of the stochastic optimisation-based scheduling method. Additionally, because Pareto fronts of non-dominated solutions are generated by SA, a suitable selection criterion is required to identify a representative solution from the archive. This step is considered essential, particularly in real-world scheduling contexts where a single actionable plan is required by decision-makers.

As the solution selection criterion, a weighted sum of the objectives F_1 and f_3 was used. The weight for F_1 was set to 0.8, and for f_3 , it was defined as $0.2 \times (n_{\text{cases}}/n_{\text{sessions}})/5$, with both values normalised before applying the selection. The choice of weights was guided by observations from the SCIP-based experiments. Since f_3 is inversely sensitive to the number of cases, its contribution was scaled based on the case-to-session ratio. The weight for f_3 equals 0.2 when this ratio is 5. This

adaptive modification was introduced specifically for this setup, as the ratio varies significantly across steps, potentially impacting solution quality and fairness.

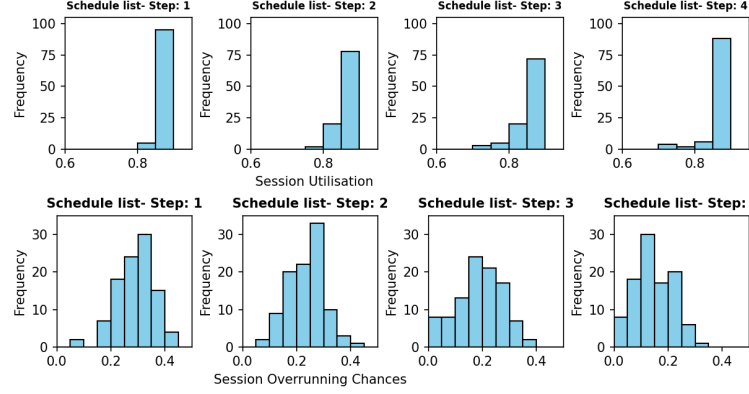


Fig. 9 Distribution of session-related KPIs at different scheduling steps, based on data collected for each session from 10 independent SA-based optimisation runs. Top: Session utilisation; Bottom: Session overrunning probability.

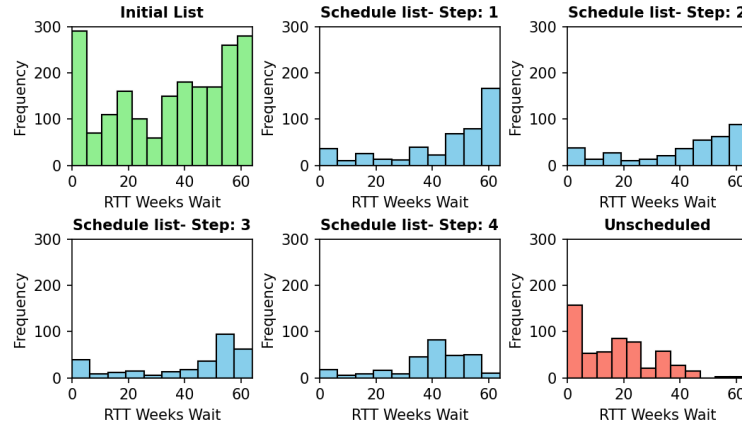


Fig. 10 RTT Weeks Waits distribution of scheduled patients at different solutions steps with data collected from 10 independent SA-based solution runs (thus data count almost 10 times than of Figure 8). Distributions for the initial patient list and the unscheduled patients are also included.

Regarding performance, while the trend from SA results is more dispersed due to the inherent variability in its stochastic search, the overall direction aligns with expectations. The session-related KPIs—such as utilisation and overrunning probability—follow a broadly similar pattern to those observed in the SCIP-based scheduling. However, the variability between runs introduces more fluctuation in the resource allocation outcomes. Session utilisation tends to be moderate across most steps (Figure 9,

top), but with less consistency in comparison to SCIP. Likewise, the overrunning probabilities are somewhat more scattered (Figure 9, bottom), reflecting the less structured optimisation path followed by SA.

When it comes to patient prioritisation, the SA-based scheduling shows slightly less sharp discrimination compared to SCIP, as shown in Figure 10. Though the RTT wait weeks distributions across the scheduling steps appear more uniform, SA does prioritise longer-waiting patients to some degree. Its performance is likely influenced by the randomness of the search process, which may not always converge on solutions that exhibit strong fairness patterns.

4.4.3 Assessing the Role of f_2 : Impact on Session Overrunning Without Overrun Control

To evaluate the necessity of incorporating the second objective (f_2), which aims to reduce the likelihood of session overruns, the step-wise scheduling experiment was repeated using the SCIP-based optimisation. In this setup, only f_1 (session utilisation) and f_3 (RTT-based patient prioritisation) were included in the objective function, while f_2 was excluded. After generating the schedules, the session overrunning probabilities were analysed post-optimisation to assess the resulting risk profile.

This focused use of SCIP was intentional. Given that SCIP requires a scalar objective function, incorporating f_2 necessitated an approximation (Section A.2) to express overrun probabilities in a linear or piecewise-linear form. Therefore, verifying the actual overrunning behaviour when f_2 is excluded also helps to validate the effectiveness of this approximation.

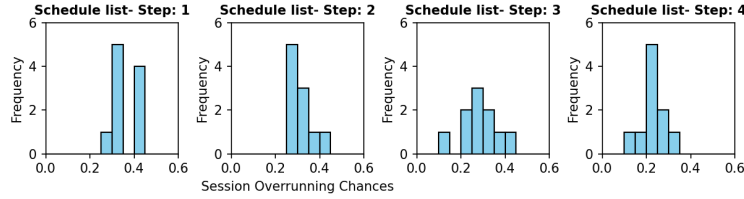


Fig. 11 Distribution of session overrunning probabilities across different scheduling steps, based on data collected from a SCIP-based optimisation run **without incorporating f_2 as an objective**.

Compared to the original runs that included f_2 (Figure 7, bottom), this change showed a shift towards higher overrunning probabilities for the sessions (Figure 11). The resulting schedules exhibited a more uneven distribution of session risks, particularly in early steps where optimisation focused on maximising utilisation without penalising overruns. In contrast, incorporating f_2 in the objective function had previously led to more balanced schedules, where overrunning probabilities were more evenly distributed and generally lower. This demonstrates that f_2 effectively enforces a trade-off between utilisation and operational feasibility.

The experiment confirms that, despite the need for simplification, f_2 plays a critical role in ensuring that generated schedules are both efficient and executable in realistic

clinical settings. The relevance of f_2 becomes especially critical when higher session utilisation limits are required, as this significantly increases the likelihood of disruptive overruns.

A similar effect is evident in the SA-based optimisation, where the inclusion of f_2 led to a narrower spread of the same (opposite to overrun) across diverse problem sizes (Figure 5).

4.4.4 Scheduling Behaviour by Solution Approach

From a scheduling perspective, we observe two distinct solution behaviours depending on the optimisation approach. Across the evaluated instances, the SCIP-based formulation typically produces more stable and sharply differentiated schedules, with higher session utilisation, stronger prioritisation of long-wait patients, and lower variability across stepwise horizons. The SA-based optimiser broadly follows the same directional trends, but—consistent with its stochastic search—its outcomes are more dispersed, with greater run-to-run variation in session-level metrics and a flatter distribution of scheduled RTT waiting times, indicating weaker (though still present) prioritisation.

Overall, for the scheduling KPIs analysed in this section, the SCIP-based approach performs best, while SA offers a flexible alternative that remains useful for larger instances or tighter runtime budgets where a fast, adaptable heuristic is preferable.

5 Conclusion and Future Work

This study set out to address the practical gap in elective surgical scheduling: how to create reliable and effective patient-focused schedules that also respect organisational resource utilisation priorities while explicitly accounting for uncertainty. To meet the need for this complex problem, this study provides a solution through an integrated framework that combines machine learning, statistical modelling, and optimisation techniques.

The proposed solution begins by using pre-trained machine learning models to estimate uncertain surgical durations. Then it quickly quantifies the risk of session disruption (such as overruns and cancellations), producing information that planners can use directly rather than treating uncertainty as an afterthought. Compared with traditional simulation-based evaluation, this risk assessment can be generated in minutes, making it feasible to use during scheduling.

Since session overruns and cancellations are common in real-world settings, the framework incorporates a statistical sub-model to assess the likelihood of such unpredictable events. Notably, estimating the probability of session overruns or cancellations using traditional simulation methods would require several hours of computation. In contrast, the proposed statistical approach delivers these evaluations in just a few minutes—marking a significant improvement in efficiency.

Using these estimates, the framework assigns patients to pre-allocated sessions under a block scheduling policy while balancing multiple aims—reflecting both Trust-level constraints and patient-centred outcomes. We tested two optimisation strategies: an exact method (SCIP) that delivers the strongest solution quality, and a simulated

annealing approach that is easier to adapt and can incorporate operationally meaningful components—like overrun risk—more naturally when strict mathematical structure becomes limiting. Tested on real ESEOC data, the framework produced high-quality schedules within minutes and supports a rolling-horizon workflow where remaining patients carry forward as new capacity becomes available, aligning with how elective services actually operate.

Looking ahead, the same structure can be extended to support more responsive decision-making, including real-time updates during the day of surgery, late cancellations, and intraoperative progress. It could also be broadened to multi-specialty planning with shared resources, and refined with surgeon-specific trends to improve realism and safety while maintaining practical runtime.

Acknowledgements. This work is an outcome of a Knowledge Transfer Partnership (KTP) between the University of Essex and East Suffolk and North Essex NHS Foundation Trust (ESNEFT)

Declarations

Funding

This work is an outcome of a Knowledge Transfer Partnership (KTP) between the University of Essex and East Suffolk and North Essex NHS Foundation Trust (ESNEFT). No additional external funding was received for this study.

Conflict of interest

The authors declare that they have no conflicts of interest.

Ethics approval and consent to participate

Not applicable. This study used only anonymised, routinely collected operational data and did not involve any direct intervention with patients.

Consent for publication

Not applicable.

Data availability

The datasets generated and/or analysed during the current study are not publicly available due to organisational data governance restrictions but are available from the corresponding author on reasonable request and with permission of ESNEFT.

Materials availability

Not applicable.

Code availability

The code used for the analyses in this study is available from the corresponding author on reasonable request.

Author contribution

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Corresponding author. The first draft of the manuscript was written by Corresponding author, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Appendix A Objective 2 - Maximising Session No Overrun Probability (f_2)

A.1 Objective Formulation - Non-linear

Minimising the probability of session overruns is equivalent to maximising the probability that a session completes within its allocated time. Since we use machine learning (ML) models to estimate surgical durations, session overrun probability is computed based on the historical accuracy of ML predictions.

A.1.1 Session Duration Estimation

The total estimated duration for any session s_j (T_j^{pred}) is given by the sum of predicted surgery times for all assigned patients:

$$T_j^{\text{pred}} = \sum_{i=1}^n (t_{ij} + \gamma_{ij}) x_{ij} \quad \forall j = 1, \dots, m \quad (\text{A1})$$

where:

- n : total number of patients,
- m : total number of session blocks,
- t_{ij} : Predicted duration of surgery for patient p_i in session s_j using trained ML models (average).
- γ_{ij} : Additional setup and transition time associated with patient p_i in session s_j .
- $x_{ij} \in \{0, 1\}$: Binary decision variable indicating whether patient p_i is scheduled in session s_j .

A.1.2 Modelling Uncertainty in Predictions

Since ML-based predictions are subject to error, the actual surgery durations vary according to a relative error distribution. Let's say the relative error is represented as

$$RE = \frac{t^{\text{act}} - t^{\text{pred}}}{t^{\text{act}}}$$

which can also be expressed as

$$\frac{S - 1}{S}$$

given a scaling factor S equivalent to the error-related ratio $\frac{t^{\text{act}}}{t^{\text{pred}}}$. To model this uncertainty, we estimate the relative scaling parameters μ_S and σ_S from past sample data (N).

Given a patient p_i scheduled in session s_j , the predicted surgery duration t_{ij} is subject to estimation error. Assuming that past errors are representative of future cases, the actual duration for patient p_i in session s_j can be expressed as:

$$t_{ij}^{\text{actual}} = \mu_{S_{ij}} \cdot t_{ij} + \varepsilon_{ij} \quad (\text{A2})$$

where ε_{ij} represents random deviation from the mean, modelled as:

$$\varepsilon_{ij} \sim \mathcal{N}(0, (\sigma_{S_{ij}} t_{ij})^2) \quad (\text{A3})$$

For this, we need to compute the related scaling parameters $\mu_{S_{ij}}$ and $\sigma_{S_{ij}}$ from data from previous related samples (n), preferably using procedures related to the patient p_i performed by the same consultant in session s_j .

If an adequate number of data points related to the procedure or consultant is not available in the past data, the requirement is fulfilled by matching at least one criterion. Considering this, we chose the relative error-based approach rather than time-based sampling, since the actual procedure time may vary from a few minutes to hundreds of minutes depending on the type of procedure within the specialty.

The mean of the scaling factor ($\mu_{S_{ij}}$) and the standard deviation ($\sigma_{S_{ij}}$) of the N -samples are calculated as follows:

$$\begin{aligned} \text{Mean scaling factor: } \mu_{S_{ij}} &= \frac{1}{N} \sum_{k=1}^N \frac{t_{ijk}^{\text{act}}}{t_{ijk}} \\ \text{Standard Deviation: } \sigma_{S_{ij}} &= \sqrt{\frac{1}{N-1} \sum_{k=1}^N \left(\frac{t_{ijk}^{\text{act}}}{t_{ijk}} - \mu_{S_{ij}} \right)^2} \end{aligned}$$

where t_{ijk}^{act} represents the actual duration recorded for the k^{th} past case of a similar surgery and t_{ijk} represents predicted duration if allow the models to predict the time for the same cases.

A.1.3 Session Duration Distribution

Since individual surgery durations are modelled as normally distributed, the total session duration T_j^{actual} also follows a normal distribution:

$$T_j^{\text{actual}} \sim \mathcal{N} \left(\sum_{i=1}^n (\mu_{S_{ij}} t_{ij} + \gamma_{ij}) x_{ij}, \sum_{i=1}^n (\sigma_{S_{ij}} t_{ij})^2 x_{ij} \right) \quad \forall j = 1, \dots, m \quad (\text{A4})$$

where:

- $\mu_{T_j} = \sum_{i=1}^n (\mu_{S_{ij}} t_{ij} + \gamma_{ij}) x_{ij}$: Expected total session-time duration.
- $\sigma_{T_j}^2 = \sum_{i=1}^n (\sigma_{S_{ij}} t_{ij})^2 x_{ij}$: Variance of session duration.

A.1.4 Probability of No Overrun

Given a session time limit T_j , the probability that the session does not exceed its allocated time is computed using the cumulative distribution function (CDF) of the normal distribution:

$$P(\text{No Overrun}_j) = P(T_j^{\text{actual}} \leq T_j) = \Phi\left(\frac{T_j - \mu_{T_j}}{\sigma_{T_j}}\right) \quad (\text{A5})$$

where $\Phi(\cdot)$ is the CDF of the standard normal distribution.

A.1.5 Objective Function Formulation

To maximise the probability of no session overruns across all sessions, we define the objective function as average chances of no overrun:

$$f_2 = \frac{1}{s} \sum_{j=1}^m \Phi\left(\frac{T_j - \mu_{T_j}}{\sigma_{T_j}}\right) \quad (\text{A6})$$

This formulation ensures that surgery sessions are scheduled in a way that minimises the risk of overruns while accounting for historical estimation accuracy and prediction uncertainty.

A.2 Linearisation of f_2

It can be noted that the f_2 is nonlinear (Equation A6). To make the objective function compatible with the linear solver SCIP, $\Phi(z)$ is approximated using a linear function over a suitable domain (e.g., $z \in [-3, 3]$):

$$\Phi(z) \approx az + b \quad (\text{A7})$$

For example, using a linear approximation:

$$a = 0.197, \quad b = 0.5$$

Substituting this into the original function:

$$f_2 \approx \frac{1}{s} \sum_{j=1}^m \left(a \cdot \frac{T_j - \mu_{T_j}}{\sigma_{T_j}} + b \right) \quad (\text{A8})$$

To make the entire function linear, the nonlinear denominator σ_{T_j} is dropped, and the entire function is divided by T_j to normalise (0,1). Then the approximation becomes:

$$f_2^{\text{linear}} = \frac{a}{s} \sum_{j=1}^m \left(1 - \frac{\mu_{T_j}}{T_j} \right) + \frac{b}{s \cdot T_j} \cdot s \quad (\text{A9})$$

Since $\frac{b}{s \cdot T_j} \cdot s$ remains as a constant and does not affect optimisation in a maximisation problem, it is dropped. Also, the constant term a can be dropped. Then, the final linear objective function becomes:

$$f_2^{\text{linear}} = \frac{1}{s} \sum_{j=1}^m \left(1 - \frac{1}{T_j} \sum_{i=1}^n (\mu_{S_{ij}} t_{ij} + \gamma_{ij}) x_{ij} \right) \quad (\text{A10})$$

References

- [1] Jain, A., Dai T, B.K., Myers, C.: Covid-19 created an elective surgery backlog: how can hospitals get back on track. *Harvard Business Review* **10**(1) (2020)
- [2] Farrow, L., Gardner, W.T., Tang, C.C., Low, R., Forget, P., Ashcroft, G.P.: Impact of covid-19 on opioid use in those awaiting hip and knee arthroplasty: a retrospective cohort study. *BMJ Quality & Safety* **32**(8), 479–484 (2023)
- [3] Childers, C.P., Maggard-Gibbons, M.: Understanding costs of care in the operating room. *JAMA surgery* **153**(4), 176233–176233 (2018)
- [4] Abdelrasol, Z., Harraz, N., Eltawil, A.: Operating room scheduling problems: a survey and a proposed solution framework. In: *Transactions on Engineering Technologies: Special Issue of the World Congress on Engineering and Computer Science 2013*, pp. 717–731 (2014). Springer
- [5] Zhu, S., Fan, W., Yang, S., Pei, J., Pardalos, P.M.: Operating room planning and surgical case scheduling: a review of literature. *Journal of Combinatorial Optimization* **37**, 757–805 (2019)
- [6] Guerriero, F., Guido, R.: Operational research in the management of the operating theatre: a survey. *Health care management science* **14**, 89–114 (2011)
- [7] Koushan, M., Wood, L.C., Greatbanks, R.: Evaluating factors associated with the cancellation and delay of elective surgical procedures: a systematic review. *International Journal for Quality in Health Care* **33**(2), 092 (2021)
- [8] Ala, A., Chen, F.: Appointment scheduling problem in complexity systems of the healthcare services: A comprehensive review. *Journal of Healthcare Engineering* **2022** (2022)
- [9] Samudra, M., Van Riet, C., Demeulemeester, E., Cardoen, B., Vansteenkiste, N., Rademakers, F.E.: Scheduling operating rooms: achievements, challenges and pitfalls. *Journal of scheduling* **19**, 493–525 (2016)
- [10] Harris, S., Claudio, D.: Current trends in operating room scheduling 2015 to 2020: a literature review. In: *Operations Research Forum*, vol. 3, p. 21 (2022). Springer

- [11] Kuo, P.C., Schroeder, R.A., Mahaffey, S., Bollinger, R.R.: Optimization of operating room allocation using linear programming techniques. *Journal of the American College of Surgeons* **197**(6), 889–895 (2003)
- [12] Farsi, A., Torabi, S.A., Mokhtarzadeh, M.: Integrated surgery scheduling by constraint programming and meta-heuristics. *International Journal of Management Science and Engineering Management* **18**(4), 292–304 (2023)
- [13] Rahimi, I., Gandomi, A.H.: A comprehensive review and analysis of operating room and surgery scheduling. *Archives of Computational Methods in Engineering* **28**(3), 1667–1688 (2021)
- [14] Guido, R., Conforti, D.: A hybrid genetic approach for solving an integrated multi-objective operating room planning and scheduling problem. *Computers & Operations Research* **87**, 270–282 (2017)
- [15] Koppka, L., Wiesche, L., Schacht, M., Werners, B.: Optimal distribution of operating hours over operating rooms using probabilities. *European Journal of Operational Research* **267**(3), 1156–1171 (2018)
- [16] Cardoen, B., Demeulemeester, E., Beliën, J.: Operating room planning and scheduling: A literature review. *European journal of operational research* **201**(3), 921–932 (2010)
- [17] Lu, Q., Zhu, X., Wei, D., Bai, K., Gao, J., Zhang, R.: Multi-phase and integrated multi-objective cyclic operating room scheduling based on an improved nsga-ii approach. *Symmetry* **11**(5), 599 (2019)
- [18] Jones, D.: A practical weight sensitivity algorithm for goal and multiple objective programming. *European Journal of Operational Research* **213**(1), 238–245 (2011)
- [19] Penn, M.L., Potts, C.N., Harper, P.R.: Multiple criteria mixed-integer programming for incorporating multiple factors into the development of master operating theatre timetables. *European Journal of Operational Research* **262**(1), 194–206 (2017)
- [20] Gür, Ş., Eren, T.: Application of operational research techniques in operating room scheduling problems: literature overview. *Journal of Healthcare Engineering* **2018**(1), 5341394 (2018)
- [21] Addis, B., Carello, G., Tànfani, E.: A robust optimization approach for the operating room planning problem with uncertain surgery duration. In: *Proceedings of the International Conference on Health Care Systems Engineering*, pp. 175–189 (2014). Springer
- [22] Pulido, R., Aguirre, A.M., Ortega-Mier, M., García-Sánchez, Á., Méndez, C.A.:

- Managing daily surgery schedules in a teaching hospital: a mixed-integer optimization approach. *BMC health services research* **14**, 1–13 (2014)
- [23] Younespour, M., Atighehchian, A., Kianfar, K., Esfahani, E.T.: Using mixed integer programming and constraint programming for operating rooms scheduling with modified block strategy. *Operations Research for Health Care* **23**, 100220 (2019)
 - [24] Abdeljaouad, M.A., Bahroun, Z., Saadani, N.E.H., Zouari, B.: A simulated annealing for a daily operating room scheduling problem under constraints of uncertainty and setup. *INFOR: Information Systems and Operational Research* **58**(3), 456–477 (2020)
 - [25] Steins, K., Persson, F., Holmer, M.: Increasing utilization in a hospital operating department using simulation modeling. *Simulation* **86**(8-9), 463–480 (2010)
 - [26] Kamran, M.A., Karimi, B., Dellaert, N.: Uncertainty in advance scheduling problem in operating room planning. *Computers & Industrial Engineering* **126**, 252–268 (2018)
 - [27] Wang, Y., Tang, J., Qu, G.: A genetic algorithm for solving patient-priority-based elective surgery scheduling problem. In: *International Conference on Intelligent Computing for Sustainable Energy and Environment*, pp. 297–304 (2010). Springer
 - [28] Xiang, W., Li, C.: Surgery scheduling optimization considering real life constraints and comprehensive operation cost of operating room. *Technology and Health Care* **23**(5), 605–617 (2015)
 - [29] Schiele, J., Koperna, T., Brunner, J.O.: Predicting intensive care unit bed occupancy for integrated operating room scheduling via neural networks. *Naval Research Logistics (NRL)* **68**(1), 65–88 (2021)
 - [30] Mosseri, J.: Predicting duration of surgery to schedule elective orthopaedic surgery. Master’s thesis, University of Toronto (Canada) (2022)
 - [31] Sapkota, M.S., Doctor, F., Herrera, H., Kampouridis, M., Yang, X.: Machine learning to predict surgery duration: Towards implementing ai and digital twin for effective scheduling. In: *2024 IEEE International Conference on Medical Artificial Intelligence (MedAI)*, pp. 435–440 (2024). IEEE
 - [32] Achterberg, T.: Scip: solving constraint integer programs. *Mathematical Programming Computation* **1**, 1–41 (2009)
 - [33] Maher, S., Miltenberger, M., Pedroso, J.P., Rehfeldt, D., Schwarz, R., Serrano, F.: PySCIPOpt: Mathematical programming in python with the SCIP optimization suite. In: *Mathematical Software – ICMS 2016*, pp. 301–307. Springer, ??? (2016).

https://doi.org/10.1007/978-3-319-42432-3_37