Healthcare Predictive Timetabling: A predictive decomposition for integrated healthcare timetabling

47 (2026) 200514

Contents lists available at ScienceDirect

Operations Research, Data Analytics and Logistics

journal homepage: https://www.sciencedirect.com/journal/operations-research-data-analytics-and-logistics

Healthcare Predictive Timetabling: A predictive decomposition for integrated healthcare timetabling☆

Daniele Giovanni Gioia [a], Lorenzo Mazza [b], Manuel Macis [b], Edoardo Fadda [b,*], Paolo Brandimarte [b]

[a] German Aerospace Center, Institute for the Protection of Terrestrial Infrastructure, Rathausallee 12, Sankt Augustin, 53757, Germany
[b] Dipartimento di Scienze Matematiche "Giuseppe Luigi Lagrange", Politecnico di Torino, Corso Duca degli Abruzzi, 24, Torino, 10129, Italy

ARTICLE INFO

Dataset link: https://github.com/DanieleGioia/HealthcarePredictiveTimetabling

Keywords:
Health care management
Timetabling
Scheduling
Model predictive control
Integrated planning

ABSTRACT

Healthcare timetabling has long been a well-studied problem in operations research. The direct influence on the efficiency of critical resources, along with the inherent complexity and interdependencies of real-world systems, has driven the development of progressively more sophisticated models and algorithms to assist decision-makers. The importance and challenges of the problem have also fostered the organization of international research competitions, the most recent of which is the Integrated Healthcare Timetabling Competition (IHTC 2024). The problem addressed in the IHTC 2024 unifies three coupled sub-problems. Specifically, the problem jointly considers Surgical Case Planning (SCP), Patient Admission Scheduling (PAS), and the Nurse-to-Room Assignment (NRA). To address the computational complexity of these three problems, as well as the time and hardware limitations imposed by the competition, this work proposes a heuristic based on a two-phase, rolling-horizon mixed-integer linear programming model. The first phase solves a reduced joint formulation of the first two subproblems, SCP and PAS, while approximating the optimal value of the NRA. The solution obtained for SCP and PAS is then fixed, and the NRA is solved in the second phase. In the first phase, the model is decomposed using a rolling-horizon strategy: as the planning window moves further into the future, constraints are progressively relaxed, and the effects of previously fixed variables are represented only approximately. Experiments are conducted on all public and hidden instances of the competition, and additional experiments under different experimental settings are included. The implementation is provided as open source.

1. Introduction

Effective resource management is crucial in healthcare. Thoughtfully optimized schedules reduce waiting times, improve coordination among providers, and enable better services. Yet meeting these high-stakes requirements is far from trivial, creating interesting operational challenges. Advanced optimization techniques have become a focal point for health-system leaders seeking smoother operations, higher patient satisfaction, and improved overall performance.

Examples are the problems related to nurse rostering [1], more general personnel scheduling [2], patient-to-room assignment [3], operating room scheduling [4], outpatient appointment scheduling [5], and many others. For a recent overview of the field, we refer the reader to [6].

Beyond narrowly defined models tied to isolated resources, integrated planning in healthcare contexts has attracted increasing interest. A comprehensive discussion of the topic is given in [7], where a wide range of studies with different assumptions are reviewed and categorized, emphasizing the importance of addressing these problems holistically. In fact, evaluating the interconnectedness of various resources in complex systems, in contrast to treating fragmented subproblems, is desired for developing better planning strategies. However, solving the integrated problem using exact methods is often computationally intractable due to the large number of variables and constraints; therefore, tailored heuristic or approximation methods are typically required to obtain practical solutions.

A pivotal example is given by the work of [8], where the assignment of patients to rooms and nurses to patients is considered jointly and it is solved by means of a sequential greedy heuristic. By contrast, a heuristic based on simulated annealing is used in [9].

The importance of the integrated perspective for healthcare applications has inspired the Integrated Healthcare Timetabling Competi-

☆ This article is part of a Special issue entitled: 'IHTP' published in Operations Research, Data Analytics and Logistics.

* Corresponding author.
E-mail addresses: daniele.gioia@dlr.de (D.G. Gioia), lorenzo.mazza@polito.it (L. Mazza), manuel.macis@studenti.polito.it (M. Macis), edoardo.fadda@polito.it (E. Fadda), paolo.brandimarte@polito.it (P. Brandimarte).

https://doi.org/10.1016/j.ordal.2026.200514
Received 19 July 2025; Received in revised form 30 June 2026; Accepted 12 August 2026
Available online 19 August 2026

tion [10] (IHTC-2024). The competition follows a series of healthcare-related events such as the first and second international nurse roster competitions [11,12]. The problem concerns a deterministic case study that must simultaneously tackle three problems: *Surgical Case Planning* (SCP), *Patient Admission Scheduling* (PAS), and *Nurse-to-Room Assignment* (NRA). A detailed and complete problem specification is given in [13]. However, to keep the presentation of this work self-contained, a summary of the specifics of the problem is included in Section 3.

The solution method proposed in this manuscript employs a *rolling-horizon* approach to decompose the integrated optimization problem and make the resulting high-dimensional models tractable. At each time step, a reduced-complexity look-ahead model is solved using an approximation of the remaining part of the problem; decisions for the current step are implemented, and the horizon is rolled forward. The procedure exploits multi-level constraint relaxation and progressive refinement. As the look-ahead extends deeper into the future, constraints are gradually relaxed in the reduced problem and then iteratively tightened while maintaining feasibility. Thus, the method preserves the predictive, iterative essence of look-ahead policies. Still, its primary purpose is computational efficiency and decomposition for large-scale optimization rather than classical control of uncertain systems. This logic has been successfully applied across a range of domains (see, e.g., [14] for direct look-ahead policies and [15] for a model predictive control perspective), and formal examples of multi-level computational advantages also appear in the control literature (e.g., [16]).

This paper reports the solution strategy adopted by the team *Health Sabauda* for the IHTC-2024. Specifically, Section 2 positions the work within the literature. Section 3 describes the overall problem. In Section 4, two Mixed Integer Linear Programming (MILP) models for the joint SCP-PAS and NRA problems are introduced, while Section 5 presents the fundamental steps to achieve an efficient solution strategy with a model predictive scheme for decomposition and an objective function approximation. Computational details and results are described in Section 6. Finally, Section 7 concludes the article by summarizing the findings and outlining potential directions for future research.

2. Literature review

In this section, we review healthcare timetabling studies that use rolling-horizon methods to relax constraints concerning future decisions within each subproblem. To the best of our knowledge, this framework has not yet been used for the integrated problem addressed in the challenge. Since the problem addressed in this paper integrates SCP, PAS, and NRA, we focus on works addressing these three problems.

We organize the review by distinguishing two motivations for adopting a rolling-horizon strategy: (i) improving the algorithmic performance on a *deterministic* problem, where all data are known in advance but the problem is too large or complex to be solved monolithically, and (ii) coping with *uncertainty*, where data become available gradually over time and the horizon is rolled forward to incorporate new information. The approach proposed in this paper belongs to the first category. However, since contributions in the deterministic setting are scarce in the healthcare scheduling literature, we also survey the more abundant stochastic stream to provide a comprehensive picture.

Regarding the SCP problem, [17] address an integrated multi-appointment blueprint scheduling problem at an orthopedic department. They propose a three-phase matheuristic whose first phase is a rolling-horizon heuristic that constructs an initial solution by sequentially solving subproblems over shorter time windows. The problem is deterministic, as all patient queues and resource availabilities are known, and the rolling horizon is adopted to handle the size and complexity of the integrated formulation, yielding solutions approximately 6% better than commercial solvers on large instances.

In the stochastic domain, [18] propose a rolling-horizon MILP for daily surgery blocks. They compare a one-shot schedule and a rolling schedule that is updated daily with new patient requests. Completed surgeries are removed and new requests are added each day. Their results show that the rolling-horizon model significantly improves operating room utilization. On a longer horizon, [19] schedule elective surgeries over multiple weeks using a rolling scheme. They formulate an integer linear program for patient selection and assignment to weekly operating room blocks over an n-week look-ahead. Each week, they implement only the first week's schedule and roll the window forward. Crucially, they allow limited disruption: when unexpected delays occur, the next re-solve can slightly adjust previously made assignments. More recently, [20] use a rolling-horizon MILP that integrates the master surgical schedule (weekly blocks per specialty) and daily case allocation. Their MILP incorporates stochastic surgery durations and non-elective arrivals; solving it in a rolling fashion (e.g., one-week horizon, shift by one week each time) yields a robust plan. In practice, the first-week schedule is fixed, and the rest is re-optimized as time advances.

To our knowledge, no work in the PAS literature uses a rolling-horizon decomposition on a purely deterministic problem formulation. Instead, the rolling-horizon framework has been applied in a stochastic setting in [21], where the authors develop two MILPs: the patient flow problem with fixed admission dates and the patient flow problem with variable admission dates. In this scheme, they allow a greater flexibility in admission days for later patients, focusing on near-term admissions first. Their experiments show that the rolling-horizon version achieves better economic results than the static one due to better management of uncertainty. In [22], the authors allow a flexible planning horizon and complex "delay" costs (patient waiting penalties). While they use local-search heuristics rather than exact MILP, their model explicitly postpones non-critical constraints: patients beyond the initial days can be shifted at low penalty. By iteratively re-planning (e.g., rescheduling weekend appointments each week), they achieve high-quality schedules under dynamic arrivals. Though not explicitly labeled rolling horizon, their flexible-horizon approach embodies the same idea of deferring future assignment constraints.

Finally, in the NRA problem, the rolling-horizon strategy has been used in several works. In particular, [23] propose a branch-and-price algorithm for the nurse scheduling problem from the second international nurse rostering competition [12]. To construct initial solutions for their large-neighborhood-search framework, they employ a rolling-horizon heuristic that sequentially computes weekly schedules in chronological order. A key aspect is the treatment of future decisions: when computing the schedule of a given week, subsequent weeks are included in the formulation but their integrality constraints are relaxed, thereby reducing computational effort while preserving an estimate of future impacts. The rolling horizon is used purely to improve scalability and solution quality. On the same problem, the authors of [24] use a look-ahead within a multi-stage MILP formulation for nurse rostering under uncertainty, where demand is revealed week by week.

A rolling-horizon approach for the stochastic problem has been considered by [25], where the authors explicitly use a rolling-horizon scheme for home-health nurse scheduling with continuity-of-care constraints. Every day, they adjust existing nurse–patient assignments and solve a mixed-integer model for the next day, yielding a new schedule that retains as much continuity as possible. Their ILP model uses approximations to allow partial decisions (e.g., relaxing some future assignments) and a heuristic for speed. This "consistency" problem is solved iteratively so that each daily plan preserves parts of the previous schedule. Beyond home-care, several authors address nurse or patient–room assignments in hospitals using rolling horizons in a stochastic setting. In more detail, [26,27] study dynamic patient-to-room assignment with nurse workload considerations. They employ a fast heuristic that recomputes the assignment whenever new events (admissions/cancellations) occur. This allows them to optimize for multiple stakeholders (patients, nurses, doctors) and handle gender or equipment constraints by repeatedly re-solving as time progresses. In summary, these works all decouple near-term and future decisions: they fix immediate nurse assignments and relax longer-term constraints until the next re-optimization, thus preserving flexibility.

A closely related methodological paper, although outside the healthcare domain, is [28]. There, the authors apply a constructive matheuristic for the shift minimization personnel task scheduling problem. Their method decomposes the problem into blocks solved sequentially to optimality, extending the current subproblem with a portion of the next block in relaxed form. Their setting is fully deterministic, and the look-ahead serves purely to improve scalability and solution quality, placing it in the first category of rolling-horizon strategies.

It is worth noting that the heuristic presented in this paper generalizes to dynamic settings where uncertainties in parameters are revealed gradually, thus moving to an adaptive decision-oriented use of the look-ahead. For example, the use of rolling-horizon strategies for tackling uncertainties in operational bed-assignment problems has been studied by [27], where a forecast is used to account for changes in lengths of stay and unknown arrivals of elective and emergency patients.

3. Problem description

The Integrated Healthcare Timetabling Problem (IHTP), as defined in the Integrated Healthcare Timetabling Competition 2024, concerns a multi-resource scheduling problem over a fixed number of days, which ranges from 14 to 28, always being a multiple of 7. Each day is additionally divided into three distinct shifts (early, late, and night) for nurse scheduling. The challenge consists of optimizing hospital operations by jointly solving three interrelated subproblems:

- Patient Admission Scheduling (PAS): deciding admission dates (coincident with the day of surgery) and room assignments.
- Surgical Case Planning (SCP): assigning surgeries to operating theaters (OTs).
- Nurse-to-Room Assignment (NRA): assigning nurses to rooms across shifts.

Each patient has attributes including mandatory/optional status, release/due dates, length of stay, gender, age group, incompatible rooms, assigned surgeon (with a fixed duration), and vectors indicating workload and required nurse skill levels across shifts.

Surgeons have a per-day surgery time limit and are tied to specific patients, but their operating theater is to be selected. All operating theaters are assumed to be identical, with capacity determined by the daily available operating time, which may vary or be unavailable on certain days.

The nurses are characterized by predetermined rosters, skill levels, and shift-based workload limits. They are assigned to rooms per shift and can be assigned to more than one room in each shift, but only one nurse is assigned to a room in a shift.

Rooms have fixed bed capacity and may be incompatible with certain patients. Moreover, a patient can only be assigned to a single room for the entire stay, and there are occupants as boundary conditions. Practically, some patients are already in the hospital at the beginning of the period and their room and stay are fixed; they impact room/nurse constraints, but not the operating theaters. The constraints of the problem are listed as reported in [13], with (**H**) labels representing hard constraints that cannot be violated, while (**S**) represent constraints that directly contribute to the objective function as weighted penalties.

H1 Gender mix in rooms is not possible

H2 Room–patient compatibility must be respected

H3 Surgeon daily capacity limits must be respected

H4 Operating theater daily capacity limits must be respected

H5 Mandatory patients must be admitted within the scheduling horizon

H6 Patients can only be admitted after their release date and (if mandatory) before their due date

H7 Room capacity must be respected

S1 The maximum difference between age groups within a room should be minimized

S2 The difference between the nurse qualification and the patient needs is penalized

S3 The number of different nurses assigned to a patient should be minimized

S4 Exceeding workload of a nurse is penalized

S5 Number of operating theaters used per day should be minimized

S6 Surgeons should be assigned to the lowest possible number of different operating theaters per day

S7 Admission delay from the patient release date is penalized

S8 The number of optional patients not admitted is penalized

4. Mathematical model

In this section, we build two MILP models to address the integrated problem. The first, called SCP+PAS, addresses both *Surgical Case Planning* (SCP) and *Patient Admission Scheduling* (PAS). Its goal is to assign surgeons to operating theaters, decide the admission date of the patients, and allocate patients to rooms. Instead, the second addresses the problem *Nurse-to-Room Assignment* (NRA), which deals with the assignment, for each shift, of nurses to occupied rooms. It is worth noting that the hard constraints of the integrated problem apply exclusively to SCP and PAS, since the NRA is always feasible. Unless otherwise specified, each decision variable is defined for all elements of the sets associated with its indices.

4.1. Surgical case planning & patient admission scheduling

Due to the size of the model, we first define sets, parameters, and variables in Section 4.1.1, constraints in Section 4.1.2, and finally the objective function in Section 4.1.3.

4.1.1. Sets, parameters, and variables

For modeling SCP+PAS, the following sets are defined:

- $\mathcal{S} = \{1, \dots, S\}$: Set of surgeons.
- $\mathcal{R} = \{1, \dots, R\}$: Set of rooms.
- $\mathcal{P} = \{1, \dots, P\}$: Set of patients; it contains the subset of mandatory patients, called $\mathcal{P}_m$. Each patient is associated with a *unique* surgeon. Thus, we call $\mathcal{P}_s$ the set of patients that have surgeon $s \in \mathcal{S}$.
- $\mathcal{O} = \{1, \dots, O\}$: Set of occupants, i.e., patients who occupy rooms (since they are already undergoing hospitalization, scheduling is not necessary).
- $\mathcal{OT} = \{1, \dots, OT\}$: Set of operating theaters.
- $\mathcal{T} = \{1, \dots, T\}$: Set of days in the planning period.
- $\mathcal{G} = \{\mathsf{M}, \mathsf{F}\}$: Set of genders.
- $\mathcal{A} = \{0, \dots, A - 1\}$: Set of age classes.

Furthermore, several parameters define the problem characteristics. Specifically:

- For each patient $p \in \mathcal{P}$, we call $\texttt{dur}[p]$ the duration of the operation in minutes, $\texttt{release\_date}[p]$ the release date (i.e., first possible availability), and $\texttt{due\_date}[p]$ the due date (only defined for mandatory patients $p \in \mathcal{P}_m$). Moreover, we call their age $\texttt{age}[p]$, the set of incompatible rooms $\texttt{incompatible\_r}[p]$, the length of stay $\texttt{len\_st}[p]$. Finally, we define a delay matrix $\texttt{del}[p,t]$ representing the delay from the release date for patient p if accepted at day t.
- $\texttt{cap\_ot}[ot,t]$ the available time for operating theater ot at day t.
- $\texttt{cap\_s}[s,t]$ the available time for surgeon s at day t.
- Two lists $\texttt{pat\_gen}[g]$ with patients for a given gender g.
- For each occupant $o \in \mathcal{O}$, their age class $\texttt{age}[o]$, and a binary matrix $\texttt{occ}^{\texttt{room}}_{o,r,t}$ whose components are 1 if occupant o is in room r at day t, 0 otherwise.
- Two lists $\texttt{occ}_g[g]$ with indexes of occupants for a given gender g.
- The capacity of room r, called $\texttt{cap\_r}[r]$.

The decision variables are:

- $\Lambda_{p,r,t} \in \{0,1\}$: 1 if patient p is in room r at day t.
- $X_{p,ot,t} \in \{0,1\}$: 1 if patient p is assigned to operating theater ot at day t.
- $Y_{ot,t} \in \{0,1\}$: 1 if operating theater ot is open at day t.
- $Z_{s,ot,t} \in \{0,1\}$: 1 if surgeon s is operating in ot at day t.
- $N_{s,t} \in \{0,\dots,OT-1\}$: Number of transfers of surgeon s at day t.
- $A^{\min}_{r,t} \in \mathcal{A}$: Minimum age group in room r at day t.
- $A^{\max}_{r,t} \in \mathcal{A}$: Maximum age group in room r at day t.
- $\delta_{p,r} \in \{0,1\}$: 1 if patient p resides in room r during their stay.
- $G_{r,t} \in \{0,1\}$: 1 if at least one patient of gender M is in room r at day t.

4.1.2. Constraints

The constraints for the SCP+PAS model are the following: Patients cannot be accepted before their release date, thus

$$X_{p,ot,t} = 0 \qquad \forall t < \texttt{release\_date}[p],\ \forall p \in \mathcal{P},\ \forall ot \in \mathcal{OT}. \tag{1}$$

A patient can only be operated once, thus

$$\sum_{ot,t} X_{p,ot,t} \le 1 \qquad \forall p \in \mathcal{P}. \tag{2}$$

The mandatory patients must undergo surgery, therefore

$$\sum_{ot} \sum_{t\in\{\texttt{release\_date}[p],\dots,\texttt{due\_date}[p]\}} X_{p,ot,t} = 1 \qquad \forall p \in \mathcal{P}_m, \tag{3}$$

where the time window represents the available days allocated for mandatory patients.

Operating theaters and surgeons have fixed capacity, thus

$$\sum_{p} \texttt{dur}[p] \cdot X_{p,ot,t} \le \texttt{cap\_ot}[ot,t] \cdot Y_{ot,t} \qquad \forall ot \in \mathcal{OT},\ \forall t \in \mathcal{T}, \tag{4}$$

$$\sum_{ot,p\in\mathcal{P}_s} \texttt{dur}[p] \cdot X_{p,ot,t} \le \texttt{cap\_s}[s,t] \qquad \forall s \in \mathcal{S},\ \forall t \in \mathcal{T}. \tag{5}$$

Furthermore, it is necessary to connect surgeons with patients and operating theaters, thus

$$\sum_{p\in\mathcal{P}_s} X_{p,ot,t} \le Z_{s,ot,t} \cdot |\mathcal{P}_s| \qquad \forall s \in \mathcal{S},\ \forall ot \in \mathcal{OT},\ \forall t \in \mathcal{T}, \tag{6}$$

$$\sum_{ot} Z_{s,ot,t} \le N_{s,t} + 1, \qquad \forall s \in \mathcal{S},\ \forall t \in \mathcal{T}, \tag{7}$$

where $|\mathcal{P}_s|$ is the number of elements in $\mathcal{P}_s$. Notice that the number of transfers is 0 in case the surgeon only operates in a single operating theater on day t.

Once operated, patients must be assigned to a room, thus X and Λ must be linked by

$$\sum_{r} \Lambda_{p,r,t} = \sum_{ot,\tau\in\mathcal{L}_{p,t}} X_{p,ot,\tau} \qquad \forall p \in \mathcal{P},\ \forall t \in \mathcal{T}, \tag{8}$$

where $\mathcal{L}_{p,t} = \{\max(1, t-\texttt{len\_st}[p]+1),\dots,t\}$. To explain constraints (8), consider a patient with $\texttt{len\_st} = 4$ operated on day $t = 2$. Given ot, then $X_{p,ot,2} = 1$. Therefore, the r.h.s. of constraints (8) will be 1 for $t = 2,\dots,2+\texttt{len\_st}[p]-1 = 2,\dots,5$, forcing $\Lambda_{p,r,t}$ to be 1 in at least one room r for the length of stay.

When assigning a room, incompatibility must be taken into account as

$$\Lambda_{p,r,t} = 0 \qquad \forall r \in \texttt{incompatible\_r}[p],\ \forall p \in \mathcal{P},\ \forall t \in \mathcal{T}. \tag{9}$$

Furthermore, a patient cannot be moved during hospitalization. Therefore, two constraints force room consistency and a single room. Namely,

$$\sum_{t} \Lambda_{p,r,t} \le \delta_{p,r} \cdot \texttt{len\_st}[p] \qquad \forall p \in \mathcal{P},\ \forall r \in \mathcal{R} \tag{10}$$

$$\sum_{r} \delta_{p,r} = \sum_{ot,t} X_{p,ot,t} \qquad \forall p \in \mathcal{P}. \tag{11}$$

Thanks to (11), $\delta_{p,r}$ can only be 1 for a specific room, maintained throughout $\texttt{len\_st}[p]$ with constraints (10) and (8). To compute the age class difference per room, given patients and occupants, the maximum and minimum classes are

$$A^{\min}_{r,t} \le A^{\max}_{r,t} \qquad \forall r \in \mathcal{R},\ \forall t \in \mathcal{T}, \tag{12}$$

$$A^{\min}_{r,t} \le (1-\texttt{occ}^{\texttt{room}}_{o,r,t}) \cdot (A-1) + \texttt{occ}^{\texttt{room}}_{o,r,t} \cdot \texttt{age}[o] \qquad \forall o \in \mathcal{O},\ \forall r \in \mathcal{R},\ \forall t \in \mathcal{T}, \tag{13}$$

$$A^{\max}_{r,t} \ge \texttt{occ}^{\texttt{room}}_{o,r,t} \cdot \texttt{age}[o] \qquad \forall o \in \mathcal{O},\ \forall r \in \mathcal{R},\ \forall t \in \mathcal{T}, \tag{14}$$

$$A^{\min}_{r,t} \le (1-\Lambda_{p,r,t}) \cdot (A-1) + \Lambda_{p,r,t} \cdot \texttt{age}[p] \qquad \forall p \in \mathcal{P},\ \forall r \in \mathcal{R},\ \forall t \in \mathcal{T}, \tag{15}$$

$$A^{\max}_{r,t} \ge \Lambda_{p,r,t} \cdot \texttt{age}[p] \qquad \forall p \in \mathcal{P},\ \forall r \in \mathcal{R},\ \forall t \in \mathcal{T}. \tag{16}$$

Specifically, the maximum age class per room must be greater than or equal to the minimum in constraints (12). Constraints (13) and, similarly, (15) bound $A^{\min}_{r,t}$ to the maximum age class (starting from 0) in the case of no assignment to room r for o and p, respectively. However, if assigned to r, $A^{\min}_{r,t}$ will be less than or equal to the age of o and p. The variables $A^{\max}_{r,t}$ follow a similar logic in constraints (14) and (16), but they are naturally bounded from below.

Next, it is not possible to mix genders within a room. Thus, patient and fixed occupant room assignments are constrained by gender. Moreover, rooms have a fixed maximum capacity. Such requirements are enforced together as

$$\sum_{p\in\texttt{pat\_gen}[\texttt{M}]} \Lambda_{p,r,t} + \sum_{o\in\texttt{occ}_g[\texttt{M}]} \texttt{occ}^{\texttt{room}}_{o,r,t} \le G_{r,t} \cdot \texttt{cap\_r}[r] \qquad \forall r \in \mathcal{R},\ \forall t \in \mathcal{T}, \tag{17}$$

$$\sum_{p\in\texttt{pat\_gen}[\texttt{F}]} \Lambda_{p,r,t} + \sum_{o\in\texttt{occ}_g[\texttt{F}]} \texttt{occ}^{\texttt{room}}_{o,r,t} \le (1-G_{r,t}) \cdot \texttt{cap\_r}[r] \qquad \forall r \in \mathcal{R},\ \forall t \in \mathcal{T}. \tag{18}$$

Constraints (17) and (18) trigger G to be 1 if at least one male is assigned to room r, ensuring only one gender is allowed per room, while imposing an upper bound on the cumulative number of patients.

4.1.3. Objective function

The parameters in the objective function are: ω_{open}, ω_{tran}, ω_{delay}, ω_{unsch}, ω_{age}. They weigh the number of operating theaters, the number of surgeon transfers, the delay time of the operation (in days), the cost of unscheduled patients, and the differences in age within the same room. The objective function, together with the constraints, follows as

$$\min \quad \begin{pmatrix} \omega_{open} \sum_{ot,t} Y_{ot,t} + \omega_{tran} \sum_{s,t} N_{s,t} + \omega_{delay} \sum_{p,ot,t} \texttt{del}[p,t] X_{p,ot,t} + \\ \omega_{unsch} (P - \sum_{p,ot,t} X_{p,ot,t}) + \omega_{age} \sum_{r,t} (A^{\max}_{r,t} - A^{\min}_{r,t}) \end{pmatrix} .$$
$$\text{s.t.} \quad (1) - (18) \tag{19}$$

4.2. Nurse-to-room assignment

Similar to SCP+PAS, the model is presented by first defining the sets, parameters, and variables in Section 4.2.1, followed by the constraints in Section 4.2.2, and finally the objective function in Section 4.2.3. NRA depends on the solution of model (19). In particular, we assume room assignments to be parameters rather than variables. More specifically, sets and parameters depend on the variables in Section 4.1.1.

4.2.1. Sets, parameters, and variables

Modeling NRA requires the following additional sets:

- $\mathcal{Q} = \{1, \dots, Q\}$: Set of shifts. Namely, there are three shifts (early, late, night) per day t, thus $Q = 3 \cdot T$.
- $\mathcal{N} = \{1, \dots, N\}$: Set of nurses.
- $\mathcal{L} = \{1, \dots, L\}$: Set of skill levels.

Furthermore, using results from SCP+PAS, the set of patients is reduced to the subset of accepted ones. Namely,

$$\bar{\mathcal{P}} = \{p : \sum_{ot,t} X_{p,ot,t} = 1\} \subseteq \mathcal{P}.$$

For convenience, since in NRA, occupants and patients are treated alike, a unified set is defined as $\mathcal{U} = \bar{\mathcal{P}} \cup \mathcal{O}$.

The parameters characterizing NRA are

- For each nurse, the skill level is defined as $\texttt{skill\_lev}[n]$.
- A binary matrix $\texttt{working\_shift}[n,q]$, whose elements take the value 1 when nurse n is assigned to shift q.
- The matrix $\texttt{work\_nurse}[n,q]$ defines the capacity of nurse n at shift q.
- The period of stay of patient $p \in \mathcal{U}$ is read as a parameter from SCP+PAS and defined as $\texttt{permanence}[p]$.
- Similarly, the room r for patient $p \in \mathcal{U}$ is $\texttt{pat\_room}[p]$.
- The length of stay $\texttt{len\_st}[p]$ is, for convenience, redefined in shifts.
- $\texttt{non\_empty}[r,q]$ is a binary matrix which is 1 when, at shift q, room r has at least one patient.
- The workload of a room r at shift q is $\texttt{work\_room}[r,q]$.
- An additional matrix $\texttt{pat\_count}[r,l,q]$ defines the number of patients in room r with skill level l at shift q.

The decision variables of NRA are:

- $\Gamma_{n,r,q} \in \{0,1\}$: 1 if nurse n is assigned to room r during shift q.
- $V_{n,q} \in \mathbb{R}_+$: Difference between the required skill level of the rooms and the skill level of the nurse n assigned to them at shift q.
- $W_{n,q} \in \mathbb{R}_+$: Difference between the rooms' workload and the maximum workload of nurse n assigned to them at shift q.
- $C_{n,p} \in \{0,1\}$: 1 if nurse n has been assigned to patient p.

4.2.2. Constraints

Every room can only have a single nurse and empty rooms do not need nurses, thus

$$\sum_n \Gamma_{n,r,q} \cdot \texttt{working\_shift}[n,q] = \texttt{non\_empty}[r,q] \qquad \forall r \in \mathcal{R}, \forall q \in \mathcal{Q}. \tag{20}$$

Notice that constraints (20) hold with equality since all the variables are binary.

The exceeding workload is subject to a penalty in the objective function and computed by

$$W_{n,q} \geq \sum_r \Gamma_{n,r,q} \cdot \texttt{work\_room}[r,q] - \texttt{work\_nurse}[n,q] \qquad \forall n \in \mathcal{N}, \forall q \in \mathcal{Q}. \tag{21}$$

Required skill levels are connected to nurses as

$$V_{n,q} \geq \sum_r \Gamma_{n,r,q} \cdot \texttt{mult\_skill}[n,r,q] \qquad \forall n \in \mathcal{N}, \forall q \in \mathcal{Q}, \tag{22}$$

where

$$\texttt{mult\_skill}[n,r,q] = \begin{pmatrix} \text{pat\_count}[r, \text{skill\_lev}[n], q] \\ \text{pat\_count}[r, \text{skill\_lev}[n]+1, q] \\ \vdots \\ \text{pat\_count}[r, L, q] \end{pmatrix}^T \cdot \begin{pmatrix} 0 \\ 1 \\ \vdots \\ L - \text{skill\_lev}[n] \end{pmatrix}$$

is the dot product that weighs and counts patients with skill level exceeding that of the nurse n in room r at shift q.

Variables $C_{n,p}$ are related to the $\Gamma_{n,r,q}$ by

$$C_{n,p} \cdot \frac{\texttt{len\_st}[p]}{3} \geq \sum_{q \in \texttt{permanence}[p]} \Gamma_{n,\texttt{pat\_room}[p],q} \qquad \forall n \in \mathcal{N}, \forall p \in \mathcal{U}. \tag{23}$$

Constraints (23) are big-M constraints with a tight value of M, based on the assumption that each nurse can work at most one shift per day. These constraints activate $C_{n,p}$ when nurse n is assigned to patient p.

4.2.3. Objective function

The parameters in the objective function are: ω_{skill}, ω_{work}, ω_{cont}. They weigh the skill level mismatch, the exceeding workload and the continuity of care. The objective function, together with the constraints, follows as

$$\min \quad \omega_{skill} \sum_{n,q} V_{n,q} + \omega_{work} \sum_{n,q} W_{n,q} + \omega_{cont} \sum_{n,p} C_{n,p}$$
$$\text{s.t.} \quad (20) - (23). \tag{24}$$

5. Problem decomposition and NRA approximation

The sequential scheme that first solves the *Surgical Case Planning & Patient Admission Scheduling* model (19) and subsequently the *Nurse-to-Room Assignment* model (24) guarantees feasibility for the integrated problem. In fact, only model (19) has hard constraints for the feasibility of the solution, while model (24) balances violations with penalties. Nevertheless, the formulated model suffers when dealing with a high number of variables. Given that IHTC-2024 rules allow for a maximum of 600[s] of computation time with 4 threads [13], dimensionality reduction is necessary. Specifically, analysis of the IHTC-2024 instances reveals that the main computational bottleneck is due to the SCP+PAS model (for example, in public instance i30, variable Λ would have dimension $\sim 10^5$).

To reduce the model's dimensionality, we introduce a reduced model run in a rolling horizon fashion, considering only a subset of the variables and progressively rolling the planning horizon forward by a selected amount. In particular, the rolling-horizon scheme relies on three progressively coarser temporal layers. First, a short horizon, dependent on a parameter step, over which the full model is enforced, and all hard and soft constraints are explicitly represented. Second, a feasibility horizon, dependent on a parameter H, which extends beyond step and preserves only the constraints required to guarantee feasibility, while omitting performance-related and soft constraints to limit dimensionality. Finally, the remaining part of the planning horizon is modeled in a highly aggregated form, without guarantees of feasibility, and is used solely to provide a rough anticipation of future effects. Moreover, we incorporate an approximate estimate of the NRA into the SCP+PAS model to provide a forward-looking approximation of the assignment problem and improve the overall integrated solution. Details of this approach are provided in Section 5.1. Finally, to address potential infeasibility, we develop a repair strategy, which is presented in Section 5.2.

5.1. Predictive reduction

Let clock denote the time step of the current horizon (set to 0 at the first iteration of the algorithm). The reduced model has: *(i)* all the decision variables and constraints of the full SCP+PAS for the time horizon $[\mathsf{clock}+1, \mathsf{clock}+\mathsf{step}]$; *(ii)* for all time steps in $[\mathsf{clock}+\mathsf{step}+1, \mathsf{clock}+\mathsf{H}]$, all the variables and constraints that are needed to ensure feasibility. More specifically, the medium-horizon part of the model (i.e., the one covering the interval from $\mathsf{clock}+\mathsf{step}+1$ up to $\mathsf{clock}+\mathsf{H}$) omits all soft constraints and their associated variables, such as, $A^{\max}_{r,t}$ and $A^{\min}_{r,t}$, which are consequently not defined for $t > \mathsf{clock}+\mathsf{step}$. *(iii)* For all the time steps in $[\mathsf{clock}+\mathsf{H}+1, T]$ only the variables and constraints for SCP are maintained. *(iv)* The objective function of the model includes an estimate of the value of the NRA.

By rolling this formulation forward, we gradually relax constraints (details are provided in the following) to strike a balance between computational tractability and solution quality as the planning horizon extends. For each iteration, decisions are taken *only* in the interval $[\mathsf{clock}+1, \mathsf{clock}+\mathsf{step}]$, defined as *effective*, while the horizons $\mathsf{clock}+\mathsf{H}$ and T are used as *predictive* and then discarded. In other words, we use the horizon $[\mathsf{clock}+\mathsf{step}+1, T]$ as look-ahead. The procedure is iterated by setting $\mathsf{clock} = \mathsf{clock}+\mathsf{step}$, until T is reached. This leads to computing $\left\lceil \frac{T}{\mathsf{step}} \right\rceil$ iterations. It is worth noting that the solution is guaranteed to be feasible up to time $\mathsf{clock}+\mathsf{H}$. Nevertheless, feasibility is not guaranteed over the entire horizon T (as, e.g., rooms might generate violations). If infeasibilities are introduced, the current solution is repaired, as introduced in Section 5.2. Moreover, if $\mathsf{step} = \mathsf{H} = T$, the predictive and effective time horizons coincide with the complete one, generating a complete SCP+PAS. In the following, we describe in Section 5.1.1 and in Section 5.1.2 the modifications of the constraints and of the objective function, respectively.

5.1.1. Constraints

Let us define:

- $\mathcal{T}^{\mathsf{step}} = \{\mathsf{clock}+1, \dots, \min\{T, \mathsf{clock}+\mathsf{step}\}\}$,
- $\mathcal{T}^{\mathsf{H}} = \{\mathsf{clock}+1, \dots, \min\{T, \mathsf{clock}+\mathsf{H}\}\}$.

For each step, patients assigned in previous steps (i.e., $[1, \mathsf{clock}]$) are tracked in an additional set $\mathcal{P}^{\mathsf{ass}}$; dynamically updated at rolling time. Reductions in the dimension of the decision variables are as follows:

- $\Lambda_{p,r,t}$ is only defined for $p \in \mathcal{P}^{\mathsf{red}}$ and $t \in \mathcal{T}^{\mathsf{H}}$ rather than up to T, where

  $$\mathcal{P}^{\mathsf{red}} \doteq \{p : \mathtt{release\_date}[p] \le \mathsf{clock}+\mathsf{H} \wedge p \notin \mathcal{P}^{\mathsf{ass}}\} \subseteq \mathcal{P}$$

  represents patients with a release date within H and not yet assigned during previous rolled steps.
- $Y_{ot,t}$, $Z_{s,ot,t}$, $N_{s,t}$, $A^{\min}_{r,t}$, $A^{\max}_{r,t}$, are only defined for $t \in \mathcal{T}^{\mathsf{step}}$ rather than up to T.
- $G_{r,t}$ is only defined for $t \in \mathcal{T}^{\mathsf{H}}$ rather than up to T.
- $\delta_{p,r}$ is only defined for $p \in \mathcal{P}^{\mathsf{red}}$.

Furthermore, the remaining horizon $\mathcal{T} = \{\mathsf{clock}+1, \dots, T\}$ is iteratively reduced by removing step elements at every iteration. In particular, the complete horizon length reduces, while $\mathcal{P}^{\mathsf{red}}$ enlarges due to additional availability of patients with release date from 1 to $\min\{T, \mathsf{clock}+\mathsf{H}\}$ that are not yet assigned.

Constraints (1), (2), (3), and (5) remain unchanged, besides the iterative reduction of patients and $\mathcal{T}$, as they are connected to SCP. Constraint (4) is split into two parts. Namely,

$$\sum_p \mathtt{dur}[p] \cdot X_{p,ot,t} \le \mathtt{cap\_ot}[ot,t] \cdot Y_{ot,t} \qquad \forall ot \in \mathcal{OT}, \forall t \in \mathcal{T}^{\mathsf{step}}, \tag{25}$$

$$\sum_p \mathtt{dur}[p] \cdot X_{p,ot,t} \le \mathtt{cap\_ot}[ot,t] \qquad \forall ot \in \mathcal{OT}, \forall t \in \mathcal{T} \setminus \mathcal{T}^{\mathsf{step}}. \tag{26}$$

The decision variable Y is missing in (26) as no longer defined from $\mathsf{step}+\mathsf{clock}+1$ onward, and is unnecessary for feasibility since it only contributes to valuing the number of penalties connected to operating theaters.

Given the reduced dimension of Z and N, constraints (6) and (7) are only enforced in $\mathcal{T}^{\mathsf{step}}$. The connection between Λ and X in (8), and room incompatibility in (9) are only necessary in $\mathcal{T}^{\mathsf{H}}$ and $\mathcal{P}^{\mathsf{red}}$. Also room consistency in (10) reduces in $\mathcal{T}^{\mathsf{H}}$ and $\mathcal{P}^{\mathsf{red}}$. Single room constraints (11) have no time index, but reduce from $\mathcal{P}$ to $\mathcal{P}^{\mathsf{red}}$.

Age related constraints (12), (13) and (14) are only defined in $\mathcal{T}^{\mathsf{step}}$. Furthermore, connections with Λ in (15) and (16) are only necessary in $\mathcal{P}^{\mathsf{red}}$ rather than $\mathcal{P}$. Nevertheless, as we have assigned patients in $\mathcal{P}^{\mathsf{ass}}$ from previous iterations, two new constraints are necessary. Namely,

$$A^{\min}_{r,t} \le (1 - \mathtt{ass}^{\mathtt{r}}_{p,r,t}) \cdot (A-1) + \mathtt{ass}^{\mathtt{r}}_{p,r,t} \cdot \mathtt{age}[p] \qquad \forall p \in \mathcal{P}^{\mathsf{ass}}, \forall r \in \mathcal{R}, \forall t \in \mathcal{T}^{\mathsf{step}}, \tag{27}$$

$$A^{\max}_{r,t} \ge \mathtt{ass}^{\mathtt{r}}_{p,r,t} \cdot \mathtt{age}[p] \qquad \forall p \in \mathcal{P}^{\mathsf{ass}}, \forall r \in \mathcal{R}, \forall t \in \mathcal{T}^{\mathsf{step}}, \tag{28}$$

where $\mathtt{ass}^{\mathtt{r}}_{p,r,t}$ is a binary matrix with value 1 if room r was assigned to patient $p \in \mathcal{P}^{\mathsf{ass}}$ at time t. Such a matrix is read as a parameter within the step and depends on the fixed decisions of the previous iterations.

Gender constraints are considered only up to H and only for $p \in \mathcal{P}^{\mathsf{red}}$. However, patients who have been assigned during previous iterations must be considered. Thus, (18) and (17) are modified as

$$\sum_{p \in \mathtt{pat\_gen}[\mathrm{M}]} \Lambda_{p,r,t} + \sum_{o \in \mathtt{occ}_g[\mathrm{M}]} \mathtt{occ}^{\mathtt{room}}_{o,r,t} + \sum_{p \in \mathtt{ass\_gen}[\mathrm{M}]} \mathtt{ass}^{\mathtt{r}}_{p,r,t} \le G_{r,t} \cdot \mathtt{cap\_r}[r] \qquad \forall r \in \mathcal{R}, \forall t \in \mathcal{T}^{\mathsf{H}} \tag{29}$$

$$\sum_{p \in \mathtt{pat\_gen}[\mathrm{F}]} \Lambda_{p,r,t} + \sum_{o \in \mathtt{occ}_g[\mathrm{F}]} \mathtt{occ}^{\mathtt{room}}_{o,r,t} + \sum_{p \in \mathtt{ass\_gen}[\mathrm{F}]} \mathtt{ass}^{\mathtt{r}}_{p,r,t} \le (1 - G_{r,t}) \cdot \mathtt{cap\_r}[r] \qquad \forall r \in \mathcal{R}, \forall t \in \mathcal{T}^{\mathsf{H}}, \tag{30}$$

where $\mathtt{ass\_gen}[g] \subset \mathcal{P}^{\mathsf{ass}}$ is the set of assigned patients with gender $g \in \mathcal{G}$ and $\mathtt{pat\_gen}[g]$ is restricted to currently active patients, i.e., $\mathtt{pat\_gen}[g] \cap \mathcal{P}^{\mathsf{red}}$.

Lastly, some big-M constraints can be tightened. In particular, constraint (10) becomes:

$$\sum_t \Lambda_{p,r,t} \le \delta_{p,r} \cdot \min(\mathtt{len\_st}[p], \mathsf{H}) \qquad \forall p \in \mathcal{P}^{\mathsf{red}}, \forall r \in \mathcal{R}. \tag{31}$$

Nevertheless, the results provided in the following sections will bind it to H.

5.1.2. Objective function

At every iteration, the objective function (19) reduces $\mathcal{T}$ according to the internal clock. However, the structure for the terms concerning the SCP components (i.e., ω_{delay} and ω_{unsch}), is unchanged. Conversely, sums over t for elements that multiply ω_{open}, ω_{tran}, and ω_{age} will reduce to $\mathcal{T}^{\mathsf{step}}$ rather than $\mathcal{T}$.

In order to give the SCP+PAS model a simplified view of the NRA, an approximation of its costs is included in the objective function.

We first introduce the reduced equivalent of Q as Q^{step}, where shifts are considered up to step. Two new decision variables follow:

- $F_q \in \mathbb{R}^+$: cumulative amount of skill deficit of nurses during shift q.
- $D_q \in \mathbb{R}^+$: cumulative amount of excessive workload of nurses during shift q.

They are linked to the other decision variables through the constraints

$$F_q \geq \sum_{p\in\mathcal{P}^{\text{red}},r} \Lambda_{p,r,\left\lceil \frac{q}{3}\right\rceil} \cdot (\texttt{skill\_required}[p] - \texttt{skill\_nurses}[q]) \qquad \forall q \in \mathcal{Q}^{\text{step}}, \tag{32}$$

where $\texttt{skill\_required}[p]$ is the sum of all skill levels required by patient p during their stay divided by their length of stay in terms of shifts ($3 * \texttt{len\_st}[p]$) and $\texttt{skill\_nurses}[q]$ is the average skill level provided by all nurses working during shift q.

Concerning the workload, it is flattened as

$$\begin{aligned} D_q \geq & \sum_{p\in\mathcal{P}^{\text{red}},r} \Lambda_{p,r,\left\lceil \frac{q}{3}\right\rceil} \cdot \texttt{balanced\_workload\_patients}[p] \\ & - \texttt{daily\_workload\_nurses}[q] + \sum_{p\in\mathcal{P}^{\text{ass}}} \texttt{workload\_assigned}[p,q] \\ & + \sum_{o} \texttt{workload\_occupants}[o,q] \quad \forall q \in \mathcal{Q}^{\text{step}}, \end{aligned} \tag{33}$$

where $\texttt{balanced\_workload\_patients}[p]$ is the sum of all workload produced by patient p during their stay divided by their length of stay in terms of shifts ($3 * \texttt{len\_st}[p]$), $\texttt{daily\_workload\_nurses}[q]$ is the total available workload provided by all nurses working during shift q, $\texttt{workload\_assigned}[p,q]$ is the workload produced by a previously assigned patient p during shift q and $\texttt{workload\_occupants}[o,q]$ is the workload produced by occupant o during shift q.

The objective function in (19), after being reduced in time and space, is expanded to become

$$\min \begin{pmatrix} \omega_{open} \sum_{ot,t} Y_{ot,t} + \omega_{tran} \sum_{s,t} N_{s,t} + \omega_{delay} \sum_{p,ot,t} \texttt{del}[p,t] X_{p,ot,t} + \\ \omega_{unsch}(P - \sum_{p,ot,t} X_{p,ot,t}) + \omega_{age} \sum_{r,t} (A^{\max}_{r,t} - A^{\min}_{r,t}) + \\ \omega_{skill} \sum_{q} F_q + \omega_{work} \sum_{q} D_q \end{pmatrix} \tag{34}$$

This approximation greatly simplifies the complexity of the NRA model as described in Section 4.2, but still tries to trace the general impact that decisions at the SCP+PAS level have on its costs. While the continuity of care cost contribution is completely omitted from the approximation, the skill level mismatch and exceeding workload are computed disregarding the nurses' assignment to rooms and greatly reducing the number and dimensionality of decision variables. Nevertheless, this additional approximation can be substantially improved and has not been particularly effective for the IHTC-2024 instances, yielding only marginal improvements.

5.2. Solution repair

The presented decomposition method does not guarantee the feasibility of the solution. In fact, when $\mathsf{H} < T$, decisions fixed in the past can generate conditions that prevent feasibility after the rolling. To cope with that, a repair strategy is implemented.

Specifically, when the solver fails due to an infeasibility, the clock is set to $\mathsf{clock} - \mathsf{step}$, while H is increased by the same amount (i.e., $\mathsf{H} = \mathsf{H} + \mathsf{step}$). This allows the solver to remake the decisions taken at the previous iteration, while considering the infeasibility zone. The strategy works recursively and, assuming infinite computational time, always grants a feasible solution. In fact, in the worst case, clock is reduced until 0, while H is increased up to T (thus considering feasible constraints up to the end of the time horizon). Nevertheless, we do not expect such extreme cases to occur, as even small reductions in clock are often sufficient to ensure the critical resources are used effectively during bottleneck periods.

Besides failures due to the model infeasibility, it is possible that step and H are too large to provide a model solvable within specific time constraints. In general, the time limit per decomposed model is assigned adaptively starting from the overall available time (called $\texttt{time\_tot}$). From this, at least $\texttt{time}^{\%}_{\texttt{NRA}} \cdot \texttt{time\_tot}$ seconds are reserved to NRA and the time limit of each sub-model of SCP+PAS will have an upper bound equal to

$$\frac{\texttt{time\_left}}{\left\lceil \frac{T - \mathsf{clock}}{\mathsf{step}} \right\rceil}, \tag{35}$$

where $\texttt{time\_left} = (1 - \texttt{time}^{\%}_{\texttt{NRA}}) \cdot \texttt{time\_tot}$. When the solver solves the problem to optimality or reaches the time limit, the time used is subtracted from $\texttt{time\_left}$ (also considering the model-building time), and the next iterations will have at least the same amount of time. NRA will follow using the maximum between $\texttt{time}^{\%}_{\texttt{NRA}} * \texttt{time\_tot}$ and all the residual time. However, when the time limit is reached without an incumbent feasible solution, a second repair strategy is implemented and represented in Fig. 1. Specifically, step and H are decreased and the residual computational time of SCP+PAS is scaled according to the remaining iterations.

6. Implementation details and results

The computational implementation was carried out in Python 3.9, utilizing Gurobi 9.5.1[1] as the optimization solver for the mathematical models. To have a safe margin from the 600[s] time limit of IHTC, we set $\texttt{time\_tot} = 595$[s], with at least 10% reserved to NRA.

To decide the values of both step and H, a simple policy based on the number of patients $|\mathcal{P}|$ is developed. Specifically, we solve SCP+PAS with $\mathsf{step} = 8$ and $\mathsf{H} = 13$, if there are less than 80 patients; $\mathsf{step} = 4$ and $\mathsf{H} = 9$ if the number of patients is between 80 and 150; $\mathsf{step} = 2$ and $\mathsf{H} = 7$ if the number of patients is between 150 and 300; otherwise, we set $\mathsf{step} = 2$ and $\mathsf{H} = 4$. Although we explored more elaborate selection rules during development, the simple size-based policy proved to be the most functional and robust in practice; investigating more sophisticated parameter-mapping strategies remains an interesting direction for future work.

6.1. Results on IHTC-2024 instances

All experiments were run on a computer equipped with an Intel Xeon Silver 4110 CPU (2.10 GHz). The solver was executed using 4 dedicated threads. The results for both hidden and public instances of the IHTC-2024 dataset are reported in Tables 1 and 2, respectively. In particular, results are compared with the best-known cost to date (UB), following the values and nomenclature reported in [10]. The leading solutions did not emerge from a single algorithm but rather from the top performers among the 32 teams on the public benchmark instances and from ten independent runs of each of the five finalist solvers on the hidden instances. An earlier version of this solver ranked 11th out of 32 during the public phase.

6.2. Impact of time limits

In order to assess how well the algorithm performs while constrained by shorter or longer time limits, we performed further tests on the public dataset, reducing the computation time to 300[s] and to 120[s], then increasing it to 1200[s]. Instances $i01$, $i02$, $i03$, $i05$, and $i06$ are small and already solved within the smallest time limit, so they are unaffected by this test. The outcomes, represented graphically in Fig. 2(a) (and reported in Table A.3), show that the improvement from 300[s] to 600[s] is consistently more significant than the improvement

[1] In later Gurobi versions (10+), model construction for large instances was considerably slower, making the rolling methodology less efficient. Tests across various systems and configurations (including presolve options) showed no improvement, with build times up to ten times longer than in version 9.5.1.

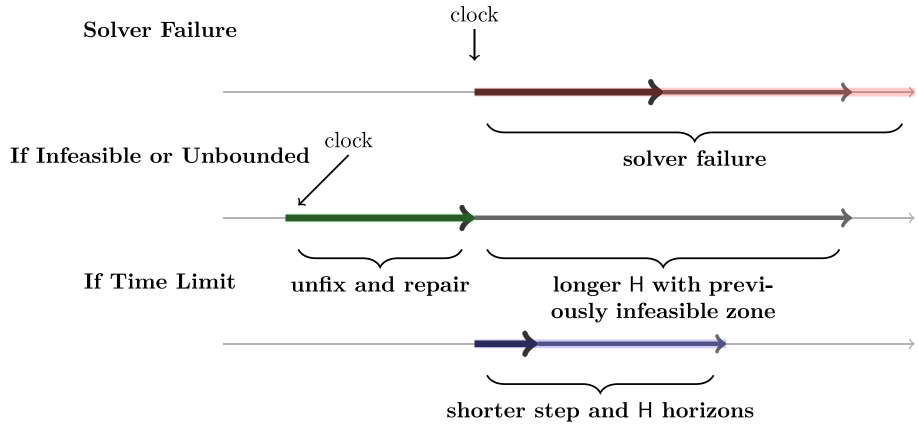

Fig. 1. Repair strategies for the solution.

Table 1
Final results - Hidden.

Instance	Cost	UB	GAP
m01	3496	3384	3.31%
m02	12 272	12 211	0.50%
m03	6839	6697	2.12%
m04	3744	3318	12.83%
m05	12 120	11 956	1.37%
m06	28 645	28 250	1.40%
m07	8973	7329	22.43%
m08	15 346	14 976	2.47%
m09	34 021	32 967	3.20%
m10	26 306	26 015	1.12%
m11	36 034	35 030	2.87%
m12	12 962	11 665	11.12%
m13	31 463	31 189	0.88%
m14	13 513	13 368	1.08%
m15	35 494	28 080	26.41%
m16	19 222	17 271	11.29%
m17	23 956	22 365	7.11%
m18	9018	7778	15.94%
m19	25 603	21 790	17.50%
m20	6020	5285	13.91%
m21	33 199	25 170	31.90%
m22	25 337	21 330	18.79%
m23	20 731	18 301	13.28%
m24	31 890	30 873	3.29%
m25	36 499	33 985	7.40%
m26	71 989	69 730	3.24%
m27	34 440	28 028	22.87%
m28	56 051	45 913	22.07%
m29	48 854	48 082	1.61%
m30	37 722	31 649	19.19%

Table 2
Final results - Public.

Instance	Cost	UB	GAP
i01	3936	3842	2.45%
i02	1416	1264	12.03%
i03	10 600	10 490	1.05%
i04	1946	1884	3.29%
i05	12 913	12 760	1.20%
i06	10 687	10 671	0.15%
i07	5576	4985	11.86%
i08	6795	6249	8.74%
i09	7608	6611	15.08%
i10	21 070	20 705	1.76%
i11	25 955	25 938	0.07%
i12	13 291	12 375	7.40%
i13	19 774	17 328	14.12%
i14	10 311	9591	7.51%
i15	13 713	12 486	9.82%
i16	11 125	10 139	9.73%
i17	48 790	40 535	20.36%
i18	37 759	37 660	0.26%
i19	51 560	43 857	17.57%
i20	31 047	29 098	6.70%
i21	28 427	24 526	15.91%
i22	62 447	47 861	30.47%
i23	39 808	37 550	6.01%
i24	33 667	33 221	1.34%
i25	13 439	11 517	16.69%
i26	74 754	64 352	16.17%
i27	67 490	50 976	32.40%
i28	76 512	75 172	1.78%
i29	20 651	12 199	69.28%
i30	40 293	37 387	7.77%

from 600[s] to 1200[s]. The gap between the cost of the proposed algorithm and the UB, obtained with 600[s], is close to the gap obtained with 1200[s], with the second-largest difference being 8.83% for instance *i*27. Instance *i*29, with a difference of 34.49%, shows the most pronounced improvement when computational time is increased. For smaller instances, the solution costs obtained under different time limits were comparable.

The results of the algorithm with a 120[s] time limit are more irregular. Compared with the runs using longer time limits, the deterioration in solution quality is limited for the smaller instances but generally becomes more pronounced for larger instances. In spite of this deterioration, and considering that lower time limits were not among the targets during development, the algorithm is still able to provide a feasible solution for 23 out of the 30 instances.

More generally, these results also reflect an inherent feature of the rolling methodology adopted in this work. Since the algorithm proceeds sequentially over multiple steps, the quality of the solution obtained at one step can influence the subsequent ones. In particular, if a step is terminated early, or if it returns a weak intermediate solution, this solution is then used to initialize the following step, and its effect may propagate through the remainder of the procedure.

6.3. Impact of parameter selection

To understand the impact that the selection of step and H has on the results, we run the algorithm with various settings. Every instance is run with each of the four possible combinations of step and H that the selection policy can choose from, evaluating its performance. This

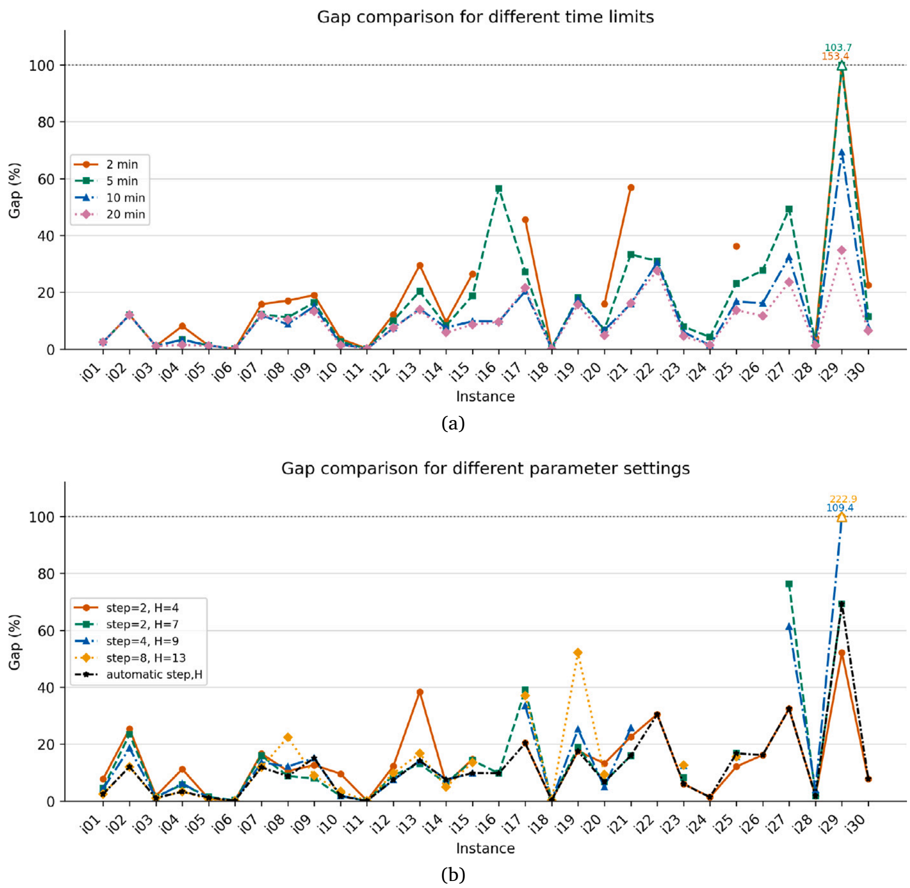

Fig. 2. Comparison of the performance of the algorithm under lower time limits and different parameter settings on the public instances.

analysis is intended to evaluate the proposed policy, rather than to perform an exhaustive grid search over all possible pairings of step and H. By analyzing the results, depicted in Fig. 2(b) (and detailed in Table A.4), we see that the selection policy of step and H consistently selects the parameters that result in the lowest gap, or one very close to it. The only exception is found in instance $i29$, where the policy underestimates the complexity of the setting and selects step = 2 and H = 7 instead of step = 2 and H = 4, thus deteriorating the solution. In fact, results for different computational times presented in Section 6.2 underlined that the quality of the solution of instance $i29$ is highly dependent on the computational time. Consequently, a smaller value for parameter H would speed up the computations, thereby improving the final solution.

6.4. Additional experiments on larger instances

We also ran our algorithm on a set of larger instances provided alongside the IHTC-2024 competition. We tested the algorithm using the standard 600[s] computational time and the extended 1200[s] one. The results, reported in Table A.5, show that for both time limits, our algorithm was able to find a feasible solution in 17 out of 24 instances.

7. Conclusions

In this paper, we present a heuristic to solve the Integrated Healthcare Timetabling problem. The method decomposes the overall problem into two subproblems, solved in a rolling horizon framework with progressive constraint relaxation and connected through an approximation.

The main advantage of this method is that it generalizes to SCP and PAS problems under any time horizon; also being naturally equipped for stochastic settings.

The method achieves an average gap of around 10% on the hidden instances and 12% on the public instances with respect to the best-known solution. Additional time-limit experiments further show that, for the proposed algorithm, increasing the computational time has little effect on the smallest instances, which are already solved within shorter limits. On the other hand, when tested on larger instances not included in the competition benchmark, the method may fail to produce feasible solutions under the imposed computational limits.

Besides refining the MILP formulation of the original problem, future studies can consider: *(i)* relaxing the integrality requirements of decision variables in the $[\mathsf{clock}+\mathsf{step}+1, \mathsf{clock}+\mathsf{H}]$ or $[\mathsf{clock}+\mathsf{H}+1, T]$ intervals, instead of completely removing them; *(ii)* the approximation of the NRA and the limited precision in its estimation within the SCP+PAS model. In fact, for the latter, better inclusion of the NRA problem in the sequential approach would allow for runs of arbitrary length, without the need for a whole-horizon nurse assignment at the end of the first stage. Concerning the parameters, although intuition suggests that larger H and step values should yield better results, this is not necessarily the case due to the complex interaction between the SCP+PAS and NRA problems.

Table A.3
Final results for different time limits on the public instances.

Inst	2 min		5 min		10 min		20 min		UB
	Cost	GAP	Cost	GAP	Cost	GAP	Cost	GAP	
i01	3937	2.47%	3936	2.45%	3936	2.45%	3936	2.45%	3842
i02	1416	12.03%	1416	12.03%	1416	12.03%	1416	12.03%	1264
i03	10 600	1.05%	10 600	1.05%	10 600	1.05%	10 600	1.05%	10 490
i04	2036	8.07%	1948	3.40%	1946	3.29%	1912	1.49%	1884
i05	12 913	1.20%	12 913	1.20%	12 913	1.20%	12 913	1.20%	12 760
i06	10 687	0.15%	10 687	0.15%	10 687	0.15%	10 687	0.15%	10 671
i07	5775	15.85%	5596	12.26%	5576	11.86%	5576	11.86%	4985
i08	7313	17.02%	6946	11.15%	6795	8.74%	6893	10.31%	6249
i09	7868	19.01%	7692	16.35%	7608	15.08%	7488	13.26%	6611
i10	21 440	3.55%	21 240	2.58%	21 070	1.76%	20 995	1.40%	20 705
i11	25 984	0.18%	25 955	0.07%	25 955	0.07%	25 955	0.07%	25 938
i12	13 875	12.12%	13 615	10.02%	13 291	7.40%	13 288	7.37%	12 375
i13	22 443	29.53%	20 845	20.30%	19 774	14.12%	19 727	13.86%	17 328
i14	10 506	9.54%	10 381	8.24%	10 311	7.51%	10 150	5.83%	9591
i15	15 789	26.46%	14 832	18.80%	13 713	9.82%	13 553	8.54%	12 486
i16	✗	–	15 873	56.56%	11 125	9.73%	11 110	9.59%	10 139
i17	58 970	45.50%	51 535	27.14%	48 790	20.36%	49 255	21.52%	40 535
i18	37 830	0.45%	37 817	0.42%	37 759	0.26%	37 716	0.15%	37 660
i19	✗	–	51 827	18.19%	51 560	17.57%	50 737	15.69%	43 857
i20	33 716	15.87%	31 093	6.85%	31 047	6.70%	30 517	4.87%	29 098
i21	38 479	56.90%	32 694	33.32%	28 427	15.91%	28 482	16.13%	24 526
i22	✗	–	62 688	30.98%	62 447	30.47%	61 093	27.63%	47 861
i23	✗	–	40 520	7.91%	39 808	6.01%	39 278	4.60%	37 550
i24	✗	–	34 627	4.23%	33 667	1.34%	33 672	1.36%	33 221
i25	15 694	36.27%	14 183	23.16%	13 439	16.69%	13 093	13.68%	11 517
i26	✗	–	82 159	27.66%	74 754	16.17%	71 875	11.69%	64 352
i27	✗	–	76 098	49.28%	67 490	32.40%	62 990	23.57%	50 976
i28	78 174	4.00%	76 698	2.03%	76 512	1.78%	76 171	1.33%	75 172
i29	30 910	153.36%	24 855	103.74%	20 651	69.28%	16 443	34.79%	12 199
i30	45 803	22.52%	41 657	11.43%	40 293	7.77%	39 797	6.45%	37 387

Table A.4
Final results for different values of step and H on the public instances.

Inst	step = 2, H = 4		step = 2, H = 7		step = 4, H = 9		step = 8, H = 13		UB
	Cost	GAP	Cost	GAP	Cost	GAP	Cost	GAP	
i01	4139	7.73%	4014	4.48%	4006	4.27%	**3936**	2.45%	3842
i02	1584	25.32%	1560	23.42%	1499	18.59%	**1416**	12.03%	1264
i03	10 675	1.76%	10 640	1.43%	10 615	1.19%	**10 600**	1.05%	10 490
i04	2095	11.20%	1993	5.78%	2000	6.15%	**1946**	3.29%	1884
i05	12 868	0.85%	12 945	1.45%	**12 913**	1.20%	12 841	0.63%	12 760
i06	10 691	0.19%	10 700	0.27%	**10 687**	0.15%	10 687	0.15%	10 671
i07	5815	16.65%	5780	15.95%	5660	13.54%	**5576**	11.86%	4985
i08	6907	10.53%	**6795**	8.74%	7007	12.13%	7649	22.42%	6249
i09	7446	12.63%	7139	7.98%	**7608**	15.08%	7203	8.95%	6611
i10	22 690	9.58%	**21 070**	1.76%	21 095	1.88%	21 405	3.38%	20 705
i11	25 971	0.13%	25 972	0.13%	**25 955**	0.07%	25 958	0.08%	25 938
i12	13 889	12.24%	13 492	9.03%	**13 291**	7.40%	13 583	9.77%	12 375
i13	23 976	38.37%	19 594	13.08%	**19 774**	14.12%	20 227	16.73%	17 328
i14	10 171	6.05%	10 190	6.25%	**10 311**	7.51%	10 059	4.88%	9591
i15	14 320	14.68%	14 269	14.27%	**13 713**	9.82%	14 186	13.60%	12 486
i16	✗	–	**11 125**	9.73%	✗	–	✗	–	10 139
i17	**48 790**	20.36%	56 365	39.05%	54 125	33.52%	55 570	37.08%	40 535
i18	37 772	0.30%	**37 759**	0.26%	37 730	0.19%	37 780	0.32%	37 660
i19	**51 560**	17.57%	52 172	18.96%	54 910	25.20%	66 717	52.15%	43 857
i20	32 936	13.19%	**31 047**	6.70%	30 520	4.89%	31 789	9.24%	29 098
i21	30 062	22.55%	**28 427**	15.91%	30 818	25.66%	✗	–	24 526
i22	**62 447**	30.47%	✗	–	✗	–	✗	–	47 861
i23	**39 808**	6.01%	40 625	8.19%	42 255	12.53%	42 300	12.65%	37 550
i24	**33 667**	1.34%	✗	–	✗	–	✗	–	33 221
i25	12 909	12.08%	**13 439**	16.69%	13 311	15.57%	13 315	15.60%	11 517
i26	**74 754**	16.17%	✗	–	✗	–	✗	–	64 352
i27	**67 490**	32.40%	89 883	76.35%	82 282	61.41%	✗	–	50 976
i28	**76 512**	1.78%	76 477	1.74%	78 159	3.98%	✗	–	75 172
i29	18 558	52.13%	**20 651**	69.28%	25 546	109.42%	39 399	222.95%	12 199
i30	**40 293**	7.77%	✗	–	✗	–	✗	–	37 387

Underlined costs correspond to the best solution among the four for each instance.
Bold costs correspond to the solution selected by our algorithm, through the parameter selection policy, for each instance.

Table A.5
Final results - Large instances.

Instance	Cost (10 min)	Cost (20 min)
l35_1	7947	7773
l35_2	30 758	30 732
l35_3	29 619	28 581
l35_4	35 129	32 296
l35_5	✗	✗
l35_6	66 819	63 566
l42_1	21 013	20 523
l42_2	14 471	14 532
l42_3	98 528	98 836
l42_4	130 998	130 337
l42_5	147 140	144 000
l42_6	110 111	104 564
l49_1	✗	✗
l49_2	97 968	97 451
l49_3	97 403	92 467
l49_4	116 782	114 776
l49_5	✗	✗
l49_6	✗	✗
l56_1	36 874	36 310
l56_2	68 934	✗
l56_3	117 253	114 840
l56_4	✗	✗
l56_5	✗	119 040
l56_6	✗	✗

CRediT authorship contribution statement

Daniele Giovanni Gioia: Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Lorenzo Mazza:** Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Manuel Macis:** Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Edoardo Fadda:** Writing – review & editing, Supervision, Software, Project administration, Methodology, Investigation, Conceptualization. **Paolo Brandimarte:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We gratefully acknowledge the organizers of the Integrated Healthcare Timetabling Competition 2024 and the participants of the 34th European Conference on Operational Research (EURO 2025) for the feedback provided during the IHTC-2024 thematic special session.

Appendix. Additional results tables

See Tables A.3–A.5.

Data availability

The code supporting the findings of this study is publicly available on GitHub at https://github.com/DanieleGioia/HealthcarePredictiveTimetabling. The problem instances of IHTC-2024 are included in the repository.

References

[1] Patrick De Causmaecker, Greet Vanden Berghe, A categorisation of nurse rostering problems, J. Sched. 14 (2011) 3–16.
[2] Jorne Van den Bergh, Jeroen Beliën, Philippe De Bruecker, Erik Demeulemeester, Liesje De Boeck, Personnel scheduling: A literature review, European J. Oper. Res. 226 (3) (2013) 367–385.
[3] Peter Demeester, Wouter Souffriau, Patrick De Causmaecker, Greet Vanden Berghe, A hybrid tabu search algorithm for automatically assigning patients to beds, Artif. Intell. Med. 48 (1) (2010) 61–70.
[4] Shuwan Zhu, Wenjuan Fan, Shanlin Yang, Jun Pei, Panos M. Pardalos, Operating room planning and surgical case scheduling: a review of literature, J. Comb. Optim. 37 (3) (2019) 757–805.
[5] Amir Ahmadi-Javid, Zahra Jalali, Kenneth J Klassen, Outpatient appointment systems in healthcare: A review of optimization studies, European J. Oper. Res. 258 (1) (2017) 3–34.
[6] Zahraa A. Abdalkareem, Amiza Amir, Mohammed Azmi Al-Betar, Phaklen Ekhan, Abdelaziz I. Hammouri, Healthcare scheduling in optimization context: A review, Health Technol. 11 (3) (2021) 445–469.
[7] Sebastian Rachuba, Melanie Reuter-Oppermann, Clemens Thielen, Integrated planning in hospitals: A review, OR Spectrum (2024) 1–54.
[8] Tabea Brandt, Tom Lorenz Klein, Melanie Reuter-Oppermann, Fabian Schäfer, Clemens Thielen, Maartje van de Vrugt, Joe Viana, Integrated patient-to-room and nurse-to-patient assignment in hospital wards, OR Spectrum (2025) 1–44.
[9] Eugenia Zanazzo, Sara Ceschia, Andrea Schaerf, Multi-neighborhood simulated annealing for the integrated patient-to-room and nurse-to-patient assignment problem, Flex. Serv. Manuf. J. (2025) 1–27.
[10] Sara Ceschia, Roberto Maria Rosati, Andrea Schaerf, Pieter Smet, Greet Vanden Berghe, Eugenia Zanazzo, The integrated healthcare timetabling competition 2024, 2024, https://ihtc2024.github.io. (Online; accessed 1 December 2025).
[11] Stefaan Haspeslagh, Patrick De Causmaecker, Andrea Schaerf, Martin Stølevik, The first international nurse rostering competition 2010, Ann. Oper. Res. 218 (2014) 221–236.
[12] Sara Ceschia, Nguyen Dang, Patrick De Causmaecker, Stefaan Haspeslagh, Andrea Schaerf, The second international nurse rostering competition, Ann. Oper. Res. 274 (1) (2019) 171–186.
[13] Sara Ceschia, Roberto Maria Rosati, Andrea Schaerf, Pieter Smet, Greet Vanden Berghe, Eugenia Zanazzo, The integrated healthcare timetabling competition 2024, Oper. Res. Data Anal. Logist. 45 (2025) 200481.
[14] Warren B Powell, Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions, John Wiley & Sons, 2022.
[15] Dimitri Bertsekas, Lessons from AlphaZero for optimal, model predictive, and adaptive control, Athena Scientific, 2022.
[16] Tobias Bäthge, Sergio Lucia, Rolf Findeisen, Exploiting models of different granularity in robust predictive control, in: 2016 IEEE 55th Conference on Decision and Control, CDC, IEEE, 2016, pp. 2763–2768.
[17] Sara Bigharaz, Henrik Andersson, Anders N. Gullhav, A rolling horizon-based matheuristic for the integrated multi-appointment blueprint scheduling problem, Comput. Ind. Eng. 212 (2025) 111700.
[18] Li Luo, Yong Luo, Yang You, Yuanjun Cheng, Yingkang Shi, Renrong Gong, A MIP model for rolling horizon surgery scheduling, J. Med. Syst. 40 (5) (2016) 127.
[19] Bernardetta Addis, Giuliana Carello, Andrea Grosso, Elena Tànfani, Operating room scheduling and rescheduling: a rolling horizon approach, Flex. Serv. Manuf. J. 28 (1–2) (2016) 206–232.
[20] Belinda Spratt, Erhan Kozan, An integrated rolling horizon approach to increase operating theatre efficiency, J. Sched. 24 (1) (2020) 3–25.
[21] Daniel Gartner, Rainer Kolisch, Scheduling the hospital-wide flow of elective patients, European J. Oper. Res. 233 (3) (2014) 689–699.
[22] Sara Ceschia, Andrea Schaerf, Dynamic patient admission scheduling with operating room constraints, flexible horizons, and patient delays, J. Sched. 19 (4) (2016) 377–389.
[23] Antoine Legrain, Jérémy Omer, Samuel Rosat, A rotation-based branch-and-price approach for the nurse scheduling problem, Math. Program. Comput. 12 (3) (2020) 417–450.
[24] Michael Römer, Taïeb Mellouli, A direct MILP approach based on state-expanded network flows and anticipation for multi-stage nurse rostering under uncertainty, in: Proceedings of the 11th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2016), Udine, Italy, 2016, pp. 549–551.
[25] Şeyma Güven-Koçak, Aliza Heching, Pınar Keskinocak, Alejandro Toriello, Continuity of care in home health care scheduling: a rolling horizon approach, J. Sched. 27 (4) (2024) 375–392.
[26] Fabian Schäfer, Manuel Walther, Alexander Hübner, Heinrich Kuhn, Operational patient-bed assignment problem in large hospital settings including overflow and uncertainty management, Flex. Serv. Manuf. J. 31 (4) (2019) 1012–1041.
[27] Fabian Schäfer, Manuel Walther, Dominik G Grimm, Alexander Hübner, Combining machine learning and optimization for the operational patient-bed assignment problem, Health Care Manag. Sci. 26 (4) (2023) 785–806.
[28] Reshma Chirayil Chandrasekharan, Pieter Smet, Tony Wauters, An automatic constructive matheuristic for the shift minimization personnel task scheduling problem, J. Heuristics 27 (2021) 205–227.