ARTICLE IN PRESS

European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Innovative applications of O.R.

A framework for integrated resource planning in surgical clinics

Thomas Reiten Bovim [a,b,*], Anders N. Gullhav [a,b], Henrik Andersson [a], Atle Riise [c]

[a] Department of Industrial Economics and Technology Management, NTNU - Norwegian University of Science and Technology, Alfred Getz vei 3, Trondheim, 7491, Norway

[b] Regional Center for Health Services Improvement, St. Olavs hospital HF, Trondheim, 7006, Norway

[c] SINTEF Digital, Department of Mathematics and Cybernetics, Forskningsveien 1, Oslo, 0373, Norway

ARTICLE INFO

Keywords:
OR in health services
Surgery scheduling
Integrated resource planning
Integer programming
Simulation

ABSTRACT

The problem under study is based on the challenges faced by the Orthopaedic Clinic at St. Olav's Hospital in Trondheim, Norway. Variations in demand and supply cause fluctuating waiting lists, and it is challenging to level the activities between the clinic's two units, the outpatient clinic and the operating theater, to obtain short waiting times for all activities. Based on these challenges, we describe and present a planning problem referred to as the Long-term Master Scheduling Problem (LMSP), where the objective is to construct an integrated Long-term Master Schedule (LMS) that facilitates short waiting times in both units. The LMS can be separated into two schedules, one cyclic high-level schedule, and one non-cyclic low-level schedule. The demand for outpatient clinic consultations and surgeries is stochastic, as are the waiting lists. To account for this, we propose a planning framework consisting of an optimization model to solve the LMSP, and a two-level planning procedure. In the planning procedure, we first solve the LMSP to construct the LMS for the upcoming planning horizon. Then, to adjust to the fluctuating waiting lists, we periodically refine the low-level schedule by solving a constrained LMSP. We also develop a simulation-based evaluation procedure to evaluate the planning framework in a real-life setting and use this to investigate different planning strategies. We find that imposing flexible, dynamic and agile planning strategies improve waiting time outcomes and patient throughput. Furthermore, combining the strategies yields additive improvements.

1. Introduction

Due to demographic changes, many Western countries experience a growing demand for health care, while the number of people in the working-age stagnates. In 2021, there were slightly more than three Europeans of working-age for every European aged 65 and above. This is a 50% higher coverage compared to 2050, when there will be fewer than two working-age adults for each elderly person (European Commission, 2023). Currently, the disruptions caused by the recent COVID-19 pandemic exert pressure on health care systems globally (OECD Publishing, 2022). From 2019 to 2020, the number of elective surgeries performed in the EU countries decreased by 16.5%, generating backlogs of patients on waiting lists. Current and future challenges put strain on the health care sector. If we want to maintain or possibly increase the level of care in the future, we must utilize the resources more efficiently.

Hulshof et al. (2012) develop a taxonomic classification of planning decisions in health care along two axes. The vertical axis reflects the hierarchical nature of decision making, including strategic, tactical and off- and online operational decisions. On the horizontal axis, the authors position major health care services, including ambulatory, emergency, surgical, inpatient, home care and residential care services. According to the classification by Hulshof et al. (2012), the decisions of interest in this work involves tactical planning within ambulatory, surgical and inpatient care services.

Tactical surgery scheduling is frequently studied in the Operations Research literature. However, tactical planning that considers the combination of the Outpatient Clinic (OC) and the operating theater (OT) is sparsely studied. Surgical patients may require services in both units, as illustrated by the stylized sketch in Fig. 1. Upon referral, all patients are put on a waiting list for an *initial* consultation (IC) in the OC. Depending on the surgeon's decision, a share of the patients require further interventions, either a *treatment* consultation (TC) in the OC or a surgery in the OT, while some patients do not require further interventions. Inpatients require a stay in a bed after surgery, while outpatients leave the hospital. Following either of these interventions, all patients require a series of *follow-up* (FU) consultations in the OC before being discharged from the system. Between each activity, patients are put on

* Corresponding author at: Department of Industrial Economics and Technology Management, NTNU - Norwegian University of Science and Technology, Alfred Getz vei 3, Trondheim, 7491, Norway.
E-mail address: thomas.bovim@ntnu.no (T.R. Bovim).

https://doi.org/10.1016/j.ejor.2024.08.021
Received 30 August 2023; Accepted 19 August 2024
Available online 22 August 2024

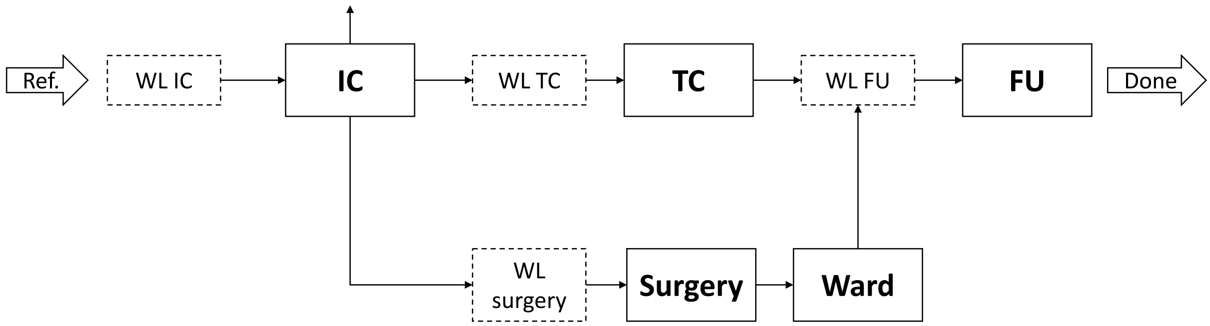


Fig. 1. The flow of patient through the OC (top) and the OT (bottom).

a waiting list (WL) for the upcoming activity, and the time between entering a waiting list and being served at the corresponding activity is referred to as the waiting time for that activity. A major objective of a surgical clinic is to obtain short waiting times across all activities.

The problem under study is based on challenges faced by the Orthopaedic Clinic at St. Olav's Hospital in Trondheim, Norway. At St. Olav's Hospital, the surgical departments (general surgery, orthopaedic surgery, neurosurgery, etc.) are decentralized and the OTs and OCs of different surgical specialties are separated in different centers. This means that the overall assignments of surgical specialties to centers are already made, and in this paper, we study the problem of assigning orthopaedic subspecialties to the orthopaedic operating rooms (ORs) and OC rooms. Variations in demand and supply cause fluctuating waiting lists, and it is challenging to level the activities between the OC and the OT to obtain short waiting times for all activities. With this as our starting point, we define and present a planning problem referred to as the Long-term Master Scheduling Problem (LMSP). The result of solving the LMSP is a Long-term Master Schedule (LMS), which covers both the OC and the OT, and that is valid for a planning horizon of typically some months. The demand considered in the LMSP is the expected activity requirements imposed both by patients already on the waiting lists and by the expected number of new arrivals. Therefore, the construction of the LMS is demand-driven, not based on serving a predefined case mix. In that sense, we take a bottom-up approach instead of a top-down approach when constructing the LMS.

The LMS can be separated in a high-level and a low-level schedule. The high-level schedule is a cyclic schedule that covers a *planning cycle* of typically one week. In the high-level schedule, surgical specialties are assigned to rooms in both units (the OC or the OT), on each day of the planning cycle. The low-level schedule is non-cyclic, and it covers all days of the planning horizon. Here, a number of surgeons of different types is assigned to each day, and a number of different activity types is assigned to one of the units on each day. Each day in the low-level schedule corresponds to a day in the high-level schedule, and the assignments made in the low-level schedule are constrained by the assignments of specialties made in the high-level schedule.

The demand for OC consultations and surgeries in the OT is stochastic, and so are the waiting lists. To account for stochastic waiting lists, we propose a two-level planning procedure where the low-level decisions are periodically refined to match the current waiting lists. We refer to the refined low-level schedule as the Refined Master Schedule (RMS).

To evaluate the proposed planning procedure in a real-life setting and under different planning strategies, we develop an evaluation procedure based on a discrete-event simulation model. In the *Flexible* strategy, a number of rooms are assigned as flexible in the high-level schedule, allowing us to postpone the assignments of specialties to these rooms until we generate the RMS. In the *Dynamic* strategy, we increase the frequency with which we construct the RMS, so that we can make more frequent adjustments according to the present waiting lists. Finally, in the *Agile* strategy, we decrease the delay between the planning and execution of the RMS, allowing us to construct the RMS with more precise information of the waiting lists.

The main contribution of this paper is a two-level planning framework composed of a formal description and mathematical formulation of the LMSP and the two-level planning procedure that accounts for stochastic waiting lists. The mathematical model extends the model developed by Bovim et al. (2022) by including explicit modeling of the waiting lists. Furthermore, we provide managerial insights related to the adoption of the different planning strategies.

In Section 2, we present relevant literature to position our work. Then, in Section 3, we describe the LMSP along with the mathematical formulation of the problem, and we introduce the two-level planning procedure. In Section 4, we describe the evaluation procedure for evaluating the planning framework, before the computational study is presented in Section 5. Finally, in Section 6, we conclude the paper.

2. Literature review and contribution

Although there is no clear definition of a Master Surgery Schedule (MSS) (Cardoen et al., 2010), it is often referred to as a cyclic schedule in which a set of surgical subgroups are assigned to OR blocks throughout the planning cycle. However, there are variations related to the decision levels and the granularity with which the subgroups are considered in the literature. As a consequence, authors also differ in the planning horizons they consider when constructing the MSS.

Some authors, such as Fügener et al. (2014) and Santos and Marques (2022), assign surgical specialties to each OR block in the MSS, and link the demand for each specialty to the case mix settled at the strategic level. These authors consider relatively long planning horizons, typically one year. Others, such as Banditori et al. (2013), Cappanera et al. (2014) and Schneider et al. (2020), propose a more frequent planning procedure and construct a new MSS every few weeks. Furthermore, these authors use a finer definition of surgical groups based on resource consumption. Both Banditori et al. (2013) and Cappanera et al. (2014) consider the current waiting lists as the demand, weakening the link to the strategic case mix decisions. Finally, some authors (Agnetis et al., 2014; Makboul et al., 2022; Mazloumian et al., 2022; Moosavi & Ebrahimnejad, 2020; Spratt & Kozan, 2016) simultaneously construct the MSS and schedule individual patients from the waiting lists. By integrating tactical and operational decisions, these authors solve the operational problem with relaxed MSS constraints. Furthermore, all these contributions consider a planning horizon of one week, implying that they study a highly dynamic and flexible system.

In this paper, we consider a two-level planning framework that covers a planning horizon of some months. We do not consider individual patients, but rather activity types based on resource requirements. First, the master schedule is constructed based on the current waiting lists and the expected future demand for services. Then, we frequently update the master schedule assignments to adjust to stochastic waiting lists.

Several authors state that OR scheduling should not be made in isolation and call for the inclusion of up- and downstream processes (Blake

& Carter, 1997; Cardoen et al., 2010; Hulshof et al., 2012). Although there exist contributions that consider the ORs in isolation (Agnetis et al., 2014; Spratt & Kozan, 2016), the vast majority of authors consider adjacent processes when constructing the MSS. The most common processes to include are downstream wards, either the Intensive Care Unit (ICU) or the medical wards. Fügener et al. (2014) build on the framework developed by Vanberkel et al. (2011) to calculate the distribution of patients resting in the downstream ward on each day of the planning cycle. However, they extend the model formulation to include an intermediate stay in an ICU before being transferred to the ward. This framework is adopted by other researchers (Fügener, 2015; Schneider et al., 2020). Although the inclusion of downstream wards is the most common, some contributions, such as Moosavi and Ebrahimnejad (2020) and Oliveira et al. (2021), also consider that some patients require a stay in a ward before surgery.

As a direction for future research, Schneider et al. (2020) propose to integrate the planning of the OC and the OT, and construct an integrated master schedule. To our knowledge, there exists only one paper that studies the integrated master scheduling of the OC and the OT (Bovim et al., 2022), and the current paper is an extension of this. First, the mathematical formulation proposed herein includes explicit modeling of waiting lists. Furthermore, the planning framework developed in this paper allows us to frequently update the schedule to account for fluctuating waiting lists.

Creating more planning flexibility in decision making demonstrates great potential (Hulshof et al., 2012). The topic of planning stability and flexibility is highly relevant in the context of designing an MSS. Here, stability refers to an MSS where all assignments are identical in each planning cycle, and thus offers predictability for the staff. Furthermore, a stable schedule allows for a more predictable pattern in terms of resource consumption. Flexibility concerns the ability to dynamically adapt the plan to the evolution of the waiting list, allowing for shorter patient waiting times. Stability and flexibility are conflicting, since the former pushes towards having a constant MSS, while the latter seeks variation if necessary. Different organizations may have different capabilities of adjusting to a changing MSS, however, they should strive to find the right trade-off between stability and flexibility (Agnetis et al., 2012). Several authors investigate the value of introducing flexibility in tactical scheduling, both related to surgery scheduling (Agnetis et al., 2012; Oliveira et al., 2021) and OC scheduling (Laan et al., 2018). They all find that introducing a very limited degree of flexibility in the master schedule will improve resource efficiency and patient waiting times. Furthermore, Agnetis et al. (2012) conclude that small but frequent changes perform better than large but infrequent changes, and Oliveira et al. (2021) find that a static, non-cyclic MSS outperforms a cyclic MSS.

In this paper, we study different planning strategies related to flexibility and dynamics. In addition, we propose an agile planning strategy, related to decreasing the delay between planning and execution of the schedule. To our knowledge, this has not been studied in the surgery scheduling literature.

3. Problem formulation

In the following, we present the Long-term Master Scheduling Problem (LMSP) along with the mathematical model. The complete mathematical model and all the notation are provided in Appendix A.

In the LMSP, we consider a surgical clinic composed of a set of units $\mathcal{U}$, with two elements in our case: the OC and the OT. The clinic serves patients from a set of surgical subspecialties (referred to as specialties in the following) $\mathcal{J}$, and the clinic's surgeons can perform a set of activity types, $\mathcal{A}$. The subset $\mathcal{A}_j^J$ includes activity types relevant for specialty j, and the activity types are either surgery types performed in the ORs in the OT, $\mathcal{A}^{OR}$, or consultation types conducted in the OC, $\mathcal{A}^{OC}$. The surgeons are of different types $\mathcal{P}$, and the subsets $\mathcal{P}_j^C$ and $\mathcal{P}_j^R$ refer to the surgeon types that are either consultants or residents, respectively, and that can serve patients of specialty j. Note that a surgeon type can serve several specialties. Consultants are more experienced than the residents, and can perform surgery types alone. Residents however, can only accompany consultants in surgeries.

A central part of the problem is to create a long-term master schedule (LMS) that covers both units for a set of days $\mathcal{D}^T$. The LMS can be separated in a high- and low-level schedule. The high-level schedule is a cyclic schedule that covers a set of cycle days, $\mathcal{D}^C$, and the main decision at this level is to assign a number of ORs and OC rooms to specialties on each cycle day. The low-level schedule is non-cyclic, covering all days of the planning horizon. Each cycle day d' corresponds to a set of days in the planning horizon, $\mathcal{D}_{d'}^T$. With a planning cycle of one week and a planning horizon of twelve weeks, each Monday in the planning horizon corresponds to the first cycle day. The main decisions at this level are to assign a number of surgeons and a number of activities of different types to each day. In Fig. 2, we illustrate an LMS covering two planning cycles for a system with two OC rooms and two ORs. The upper schedule is the cyclic high-level schedule, while the lower is the non-cyclic low-level schedule.

3.1. The high-level schedule

The opening hours of a room are referred to as a room-day, and in this problem, we only consider full-day assignments of rooms to specialties. For simplicity, we refer to the assignments of room-days as the assignments of rooms. There are two decision variables that constitute the high-level schedule of the LMS: the number of rooms assigned to specialty j in unit u on cycle day d, β_{ujd}, and the number of rooms in unit u assigned as flexible on cycle day d, y_{ud}. By assigning a room as flexible, we postpone the assignment of a specialty to this room until constructing the low-level schedule.

Constraints (1) make sure that we do not assign more rooms on cycle day d in unit u than the number of rooms available, R_u, in the unit. The number of flexible room-days are specified upfront, and constraints (2) ensure that a given number of rooms, B_u^F, are assigned as flexible in each unit. Furthermore, to account for the availability of adjacent resources, such as anesthetists, the total number of rooms that can be assigned to unit u during a cycle is limited by C_u^N, as imposed by constraints (3).

$$\sum_{j\in\mathcal{J}} \beta_{ujd} + y_{ud} \le R_u \qquad u\in\mathcal{U}, d\in\mathcal{D}^C \tag{1}$$

$$\sum_{d\in\mathcal{D}^C} y_{ud} = B_u^F \qquad u\in\mathcal{U} \tag{2}$$

$$\sum_{j\in\mathcal{J}}\sum_{d\in\mathcal{D}^C} \beta_{ujd} + \sum_{d\in\mathcal{D}^C} y_{ud} \le C_u^N \qquad u\in\mathcal{U} \tag{3}$$

The number of ORs assigned to a specialty on day d cannot exceed the number of consultants available for that specialty on that day. The number of surgeons of type p available on cycle day d is given by C_{pd}, while variable μ_{jd}^{OR} represents the remaining capacity of consultants of specialty j on day d after assigning ORs to that specialty and day. The remaining capacity can be used to access flexible ORs. Accordingly, a specialty cannot access more flexible ORs on a day than the difference between the number of consultants covering the specialty and the number of ORs assigned to that specialty on that day, which is ensured by constraints (4). Despite that some surgeon types can cover more than one specialty, a consultant cannot cover more than one specialty per day. Therefore, the total remaining capacity of consultants on day d equals the total capacity of consultants minus the total number of ORs assigned to specialties on that day. This is enforced by constraints (5). We require each flexible OR assigned on cycle day d to be flexible in the sense that more than one specialty should have the capacity to use it. If, for each specialty j', we sum the remaining capacity of all other specialties, this is an upper bound on the number of ORs that can be assigned as flexible on a day. If more ORs than the upper bound are assigned as flexible on day d, the specialty with the most remaining capacity (μ_{jd}^{OR}) will be assigned to at least one flexible OR on that day.

	Cycle 1							Cycle 2						
	M	T	W	T	F	S	S	M	T	W	T	F	S	S
OCR 1	Hand	Plastic		Arthro	Hand			Hand	Plastic		Arthro	Hand		
OCR 2	Plastic	FLEX	Arthro	Hand	Plastic			Plastic	FLEX	Arthro	Hand	Plastic		
OR 1	Arthro	Hand	Plastic	FLEX	Arthro			Arthro	Hand	Plastic	FLEX	Arthro		
OR 2	Hand	Hand	Plastic	Plastic				Hand	Hand	Plastic	Plastic			

	M	T	W	T	F	S	S	M	T	W	T	F	S	S
OCR 1	2 IC 1 TC 3 FU	3 IC 3 FU		3 IC 3 FU	2 IC 4 FU			1 IC 2 TC 3 FU	2 IC 2 TC 2 FU		1 IC 1 TC 4 FU	2 IC 1 TC 3 FU		
OCR 2	2 IC 2 TC 2 FU	(Arthro) 1 IC 5 FU	2 IC 1 TC 3 FU	3 IC 3 FU	4 IC 2 TC			2 IC 4 FU	(Hand) 3 IC 3 FU	2 TC 4 FU	3 IC 3 FU	1 IC 2 TC 3 FU		
OR 1	1 AS-2 1 AS-3	2 HS-1 1 HS-2	1 PS-2 3 PS-3	(Hand) 3 HS-3	1 AS-2 1 AS-3			3 AS-3	2 HS-1 1 HS-2	2 PS-2 2 PS-3	(Plastic) 1 PS-1 2 PS-3	2 AS-1		
OR-2	2 HS-1 1 HS-2	3 HS-3	2 PS-2 2 PS-3	2 PS-1				1 HS-1 1 HS-3	1 HS-2 1 HS-3	2 PS-2 2 PS-3	2 PS-1			

Fig. 2. An LMS covering two planning cycles for a system with two OC rooms and two ORs. Each cell represents a room-day. Top: The high-level cyclic schedule. Bottom: The low-level non-cyclic schedule. In this example, the surgery types are named according to the corresponding specialty: HS-1 refers to Hand surgery type 1, PS-3 is Plastic surgery type 3, and so on. There are three consultation activity types that can be performed in the OC: Initial consultations (IC), treatment consultations (TC) and follow-up consultations (FU).

Then, the surplus number of flexible ORs might as well be assigned to this specialty permanently. By introducing constraints (6), we make sure that each flexible OR is indeed flexible.

$$\mu^{OR}_{jd} \le \sum_{p\in\mathcal{P}^C_j} C_{pd} - \beta_{ujd} \qquad u = OT, j \in \mathcal{J}, d \in \mathcal{D}^C \qquad (4)$$

$$\sum_{j\in\mathcal{J}} \mu^{OR}_{jd} = \sum_{p\in\mathcal{P}^C} C_{pd} - \sum_{j\in\mathcal{J}} \beta_{ujd} \qquad u = OT, d \in \mathcal{D}^C \qquad (5)$$

$$\sum_{j'\in\mathcal{J}|j'\neq j} \mu^{OR}_{j'd} \ge y_{ud} \qquad u = OT, j \in \mathcal{J}, d \in \mathcal{D}^C \qquad (6)$$

3.2. The low-level schedule

There are four main decision variables that constitute the low-level schedule: the number of surgeons of type p assigned to specialty j on cycle day d, g_{pjd}, the number of rooms accessed by specialty j in unit u on day d, λ_{ujd}, the number of flexible rooms assigned to specialty j in unit u on day d, y_{ujd}, and the number of activities of type a assigned to specialty j on day d, x_{jad}.

3.2.1. Room constraints

Specialty j cannot access more rooms in unit u on day d than the number of rooms assigned to that specialty in that unit on the corresponding cycle day, which is ensured by constraints (7). Furthermore, we cannot access more flexible rooms in unit u on day d than the number of flexible rooms assigned to that unit on the corresponding cycle day, as stated in constraints (8).

$$\lambda_{ujd} \le \beta_{ujd'} + y_{ujd} \qquad u \in \mathcal{U}, j \in \mathcal{J}, d' \in \mathcal{D}^C, d \in \mathcal{D}^T_{d'} \qquad (7)$$

$$\sum_{j\in\mathcal{J}} y_{ujd} \le y_{ud'} \qquad u \in \mathcal{U}, d' \in \mathcal{D}^C, d \in \mathcal{D}^T_{d'} \qquad (8)$$

3.2.2. Surgeon constraints

We introduce a set of OR activity blocks $\mathcal{B}$, and each element b represents a combination of a number of different surgery types of a given specialty that can fit within the opening hours of the ORs. The set $\mathcal{B}^J_j$ contains the blocks that are relevant to specialty j. Parameter N^B_b is the number of surgeons required to conduct block b, and it equals the number of surgeons required in the surgery type with the highest demand for surgeons in the block. The reason for modeling blocks of surgeries is that most combinations of surgery types do not add up to one OR-day, resulting in slack in a conventional formulation. The block formulation is tighter, and we obtain a stronger bound from solving the linear relaxation of this compared to the conventional formulation. It also handles the packing of activity types, which is a challenge. Unlike the surgery types, we assume that the durations of all OC activity types are the same, and that an integer number of each OC activity type adds up to one OC room-day. As a consequence, a block formulation does not provide a tighter formulation for these activity types.

The variable x^{OR}_{bd} represents the number of blocks b assigned to day d. While some surgeries only require the presence of one surgeon, several surgeries require two surgeons. In contrast to the surgery types, all OC activity types require the presence of only one surgeon. Constraints (9) limit the mix of activity types that can be performed by specialty j in the OC and the OT on day d by limiting the surgeon requirements to the number of surgeons assigned for that specialty on that day. A consultant is required to perform surgery types, and constraints (10) limit the number of ORs that can be accessed by specialty j on day d to the number of consultants assigned to that specialty on that day. According to constraints (11), we cannot assign more surgeons of type p on day d than the number of surgeons available of that type on the corresponding cycle day. Finally, constraints (12) limit the number of days that surgeons of type p can cover during the planning horizon to C^{MAX}_p.

$$\lambda_{ujd} + \sum_{b\in\mathcal{B}^J_j} N^B_b x^{OR}_{bd} \le \sum_{p\in\mathcal{P}^C_j\cup\mathcal{P}^R_j} g_{pjd} \qquad u = OC, j \in \mathcal{J}, d \in \mathcal{D}^T \qquad (9)$$

$$\sum_{b\in\mathcal{B}^J_j} x^{OR}_{bd} \le \sum_{p\in\mathcal{P}^C_j} g_{pjd} \qquad j \in \mathcal{J}, d \in \mathcal{D}^T \qquad (10)$$

$$\sum_{j\in\mathcal{J}} g_{pjd} \le C_{pd'} \qquad p \in \mathcal{P}, d' \in \mathcal{D}^C, d \in \mathcal{D}^T_{d'} \qquad (11)$$

$$\sum_{j\in\mathcal{J}}\sum_{d\in\mathcal{D}^T} g_{pjd} \le C_p^{MAX} \qquad p\in\mathcal{P} \qquad (12)$$

3.2.3. Activity constraints

During the care process, the patients undergo a set of different activity types. After receiving an activity, patients must wait for a number of days, referred to as the activity delay, before they can receive the next activity.

The set $\mathcal{D}$ covers all days in the planning horizon and the days spanning the maximum activity delay before the planning horizon. The set $\mathcal{A}_j^{OT}$ contains all surgery types of specialty j, and the number of surgeries of type a and specialty j included in OR activity block b is given by parameter A_{bja}^B. Furthermore, parameter D_{ja}^{OC} gives the duration of OC activity type a of specialty j, while T^{OC} is the number of hours available in an OC room-day. The variable x_{jad} gives the number of activities of type a and specialty j assigned to day d. Constraints (13) ensure that we do not assign more activity blocks to specialty j on day d than the number of ORs used by that specialty on that day. The parameter X_{jad} is the number of activities of type a and specialty j planned on day d, leading up to the planning horizon. As stated in constraints (14), we must consider the activities assigned prior to the current planning horizon as these are expected to impose downstream demand in the current planning horizon. By summing over the assigned OR activity blocks, constraints (15) allow us to calculate the number of OR activities assigned for each specialty on a day. As stated by constraints (16), the duration of OC activities that can be performed by specialty j on day d cannot exceed the time staffed in the OC for that specialty on that day.

$$\sum_{b\in\mathcal{B}_j^J} x_{bd}^{OR} = \lambda_{ujd} \qquad u=OT, j\in\mathcal{J}, d\in\mathcal{D}^T \qquad (13)$$

$$x_{jad} = X_{jad} \qquad j\in\mathcal{J}, a\in\mathcal{A}, d\in\mathcal{D}|d<1 \qquad (14)$$

$$x_{jad} = \sum_{b\in\mathcal{B}_j^J} A_{bja}^B x_{bd}^{OR} \qquad j\in\mathcal{J}, a\in\mathcal{A}_j^{OT}, d\in\mathcal{D}^T \qquad (15)$$

$$\sum_{a\in\mathcal{A}^{OC}} D_{ja}^{OC} x_{jad} \le T^{OC}\lambda_{ujd} \qquad u=OC, j\in\mathcal{J}, d\in\mathcal{D}^T \qquad (16)$$

3.2.4. Ward constraints

Some surgical interventions require patients to rest in a bed for some days following surgery. There is a set of wards $\mathcal{W}$ available in the clinic, and the subset $\mathcal{W}_a^A$ specifies which wards can serve patients that received surgery of type a. Similarly, $\mathcal{A}_w^W$ is the set of activity types that can be accommodated by ward w. The scheduling of surgeries is limited by the number of beds staffed in ward w on each cycle day d, denoted A_{wd}. Our modeling of ward capacity is based on using the expected length of stay (LOS) after the different types of surgeries. We define $\mathcal{D}_{jad}^{LOS}$ as the set of days on which a patient that is resting at a ward on day d can have undergone a surgery of specialty j and surgery type a. Furthermore, we use variables u_{jawd} to represent the number of beds occupied in ward w on day d by patients of specialty j who received surgery type a. In constraints (17), we count the number of patients of surgery type a still present at the wards on day d. Constraints (18) ensure that the number of beds occupied in ward w, on day d, does not exceed the number of beds available in that ward on the corresponding cycle day.

$$\sum_{d'\in\mathcal{D}_{jad}^{LOS}} x_{jad'} = \sum_{w\in\mathcal{W}_a^A} u_{jawd} \qquad j\in\mathcal{J}, a\in\mathcal{A}^{OT}\cap\mathcal{A}_j^J, d\in\mathcal{D}^T \qquad (17)$$

$$\sum_{j\in\mathcal{J}}\sum_{a\in\mathcal{A}_w^W\cap\mathcal{A}_j^J} u_{jawd} \le A_{wd'} \qquad w\in\mathcal{W}, d'\in\mathcal{D}^C, d\in\mathcal{D}_{d'}^T \qquad (18)$$

3.2.5. Patient flow constraints

For specialty j and activity type a, there is an expected number of external patients L_{ja} entering the system each day of the planning horizon, which drives the demand for downstream activity types covered by the specialty. While we do not know beforehand what activity types a patient will require, the parameter $F_{jaa'}$ represents the probability that a patient of specialty j requires activity type a' following activity type a. The parameter D_{ja}^A represents the duration of the activity delay of activity type a and specialty j, while the expected number of patients of specialty j waiting for activity type a at the start of the planning horizon is given by parameter Q_{ja}^0. The variable q_{jad} represents the expected number of patients of specialty j waiting for activity type a on day d. Constraints (19) are valid for the first day of the planning horizon and state that the expected number of patients of specialty j waiting for activity type a, on day d, is equal to the number of patients waiting for that activity when entering the day, minus the expected number of activities of that type served on the day, and plus the expected number of activities of that type arriving on the day, either from outside the system or from the upstream activities performed one activity delay ago. Constraints (20) follow the same logic as constraints (19), but are applied to all subsequent days of the planning horizon.

$$q_{jad} = Q_{ja}^0 - x_{jad} + L_{ja} + \sum_{a'\in\mathcal{A}_j^J} F_{ja'a} x_{ja',(d-D_{ja}^A)} \qquad j\in\mathcal{J}, a\in\mathcal{A}_j^J, d=1 \qquad (19)$$

$$q_{jad} = q_{ja,(d-1)} - x_{jad} + L_{ja} + \sum_{a'\in\mathcal{A}_j^J} F_{ja'a} x_{ja',(d-D_{ja}^A)} \qquad j\in\mathcal{J}, a\in\mathcal{A}_j^J, d\in\mathcal{D}^T|d>1 \qquad (20)$$

3.2.6. Waiting list constraints

When waiting for an activity, patients are put on a waiting list for the corresponding specialty and activity type, and variable q_{jad} represents its size. The time between entering a waiting list and the time of service is referred to as the *waiting time*. We do not explicitly consider waiting time in our model, but rather use the size of the waiting lists as a means to control the waiting times. In doing this, we assume that the number of patients entering each waiting list equals the number of patients served from the waiting list over time, that is, that the queuing system is stationary.

First, we define a number of threshold waiting times for each list, separated by waiting time intervals, $\mathcal{K}$. The first interval, $k=1$ is defined from zero days to the first threshold value, while the second interval, $k=2$ is defined from the first to the second threshold value, and so on. The threshold values for specialty j and activity type a are denoted by W_{jak}. For specialty j, activity type a, and waiting time interval k, we can calculate the average number of patients on the waiting list, Q_{jak}, that corresponds to the threshold waiting time, W_{jak}, given the average arrival rate to the waiting list $\bar{\lambda}_{ja}$:

$$Q_{jak} = W_{jak}\cdot\bar{\lambda}_{ja} \qquad j\in\mathcal{J}, a\in\mathcal{A}_j^J, k\in\mathcal{K} \qquad (21)$$

Eq. (21) is referred to as Little's formula, and it applies to stationary systems. We can calculate $\bar{\lambda}_{ja}$ for all waiting lists by summing over the external and internal flow of patients entering each waiting list. Hence, for a specified system with given external arrival rates and expected flows between waiting lists, we can derive the threshold waiting lists for given threshold waiting times. Since we consider a stationary system, the waiting times and derived waiting lists (q_{jad}) represent average waiting times and waiting lists. Therefore, the waiting time thresholds should be set such that the waiting times can be controlled.

The waiting lists are monitored at specific days of the planning horizon, and the set $\mathcal{D}^Q$ specifies what days the waiting lists are measured. The variable $\bar{q}_{jadk}$ represents the number of patients on the waiting list for activity type a and specialty j that is assigned to interval k on day d. According to constraints (22), all patients on the waiting list for specialty j and activity type a are assigned to one of the intervals.

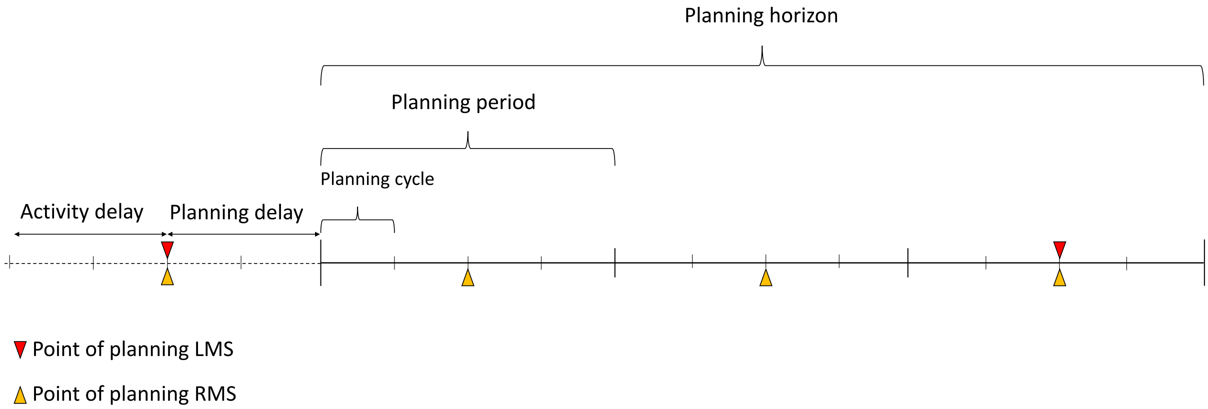


Fig. 3. Overview of a planning horizon of 84 days, consisting of three planning periods (28 days) and twelve planning cycles (7 days). The LMS is generated before the planning horizon, while the RMS is constructed before each planning period. There is a planning delay that separates the times of planning and execution of the corresponding schedules. At each time of planning, we must consider both the planning delay and the activity delay leading up to the time of planning.

Furthermore, constraints (23) make sure that we limit the number of patients on the waiting list for specialty j and activity type a that can be assigned to interval k.

$$q_{jad} = \sum_{k \in \mathcal{K}} \overline{q}_{jadk} \qquad j \in \mathcal{J}, a \in \mathcal{A}^J_j, d \in \mathcal{D}^Q \tag{22}$$

$$\overline{q}_{jadk} \leq Q_{jak} \qquad j \in \mathcal{J}, a \in \mathcal{A}^J_j, d \in \mathcal{D}^Q, k \in \mathcal{K} \tag{23}$$

3.2.7. Objective function

The objective of the LMSP is to obtain short waiting times for all activity types, and to facilitate a high throughput of patients. In the objective function, we minimize the penalty of having patients in the different intervals in all waiting lists. $\overline{C}_{jak}$ is the penalty coefficient associated with each patient of specialty j waiting for activity type a in interval k. To achieve short waiting times, $\overline{C}_{jak}$ increases with increasing intervals, and to push patients through the system, $\overline{C}_{jak}$ decreases for downstream activities. The set $\mathcal{D}^Q$ includes the days over which we measure the waiting lists.

$$\min \quad \frac{1}{|\mathcal{D}^Q|} \sum_{j \in \mathcal{J}} \sum_{a \in \mathcal{A}^J_j} \sum_{d \in \mathcal{D}^Q} \sum_{k \in \mathcal{K}} \overline{C}_{jak} \overline{q}_{jadk} \tag{24}$$

3.3. The two-level planning procedure

The demand for activity types is stochastic, and so are the waiting lists. To handle the uncertain demand, we propose a two-level planning procedure in which low-level decisions are frequently updated to adjust to the current waiting lists. We refer to the refined schedule as the Refined Master Schedule (RMS).

Fig. 3 illustrates a planning horizon of 84 days that is divided into three planning periods and 12 planning cycles. There are two red triangles included. The leftmost triangle represents the point in time when we construct the LMS for the current planning horizon, while the rightmost represents the point in time when we construct the LMS for the next planning horizon. The RMS is constructed once for each planning period, and the yellow triangles represent the points in time when the RMS is made. As we see from the figure, the LMS and RMS are both planned some days before they are executed, and we refer to the period between planning and execution as the *planning delay*. The set of days that cover the planning delay is $\mathcal{D}^{\mathcal{P}}$, and there are D^P days in the planning delay.

At the points of planning, we must calculate the expected waiting list for specialty j and activity type a at the beginning of the planning horizon (or planning period for the RMS), Q^0_{ja}, based on the current waiting list, Q^P_{ja}. We do this by considering the current waiting list, the activities already planned for in the planning delay and the expected external and internal arrivals to the waiting list. Since the current waiting list may be shorter than expected in the previous planning period, there may be less activities performed in the planning delay than we planned for, as we cannot have negative waiting lists. Eqs. (25) define how we calculate Q^0_{ja}:

$$Q^0_{ja} = \max\{0, Q^P_{ja} - \sum_{d \in \mathcal{D}^P} X_{jad} + L_{ja} D^P + \sum_{a' \in \mathcal{A}^J_j} \sum_{d \in \mathcal{D}^P} F_{ja'a} X_{ja',(d - D^A_{ja})}\} \qquad j \in \mathcal{J}, a \in \mathcal{A}^J_j \tag{25}$$

We specify three planning strategies, based on the planning framework. In the *Flexible* strategy we allow for the assignment of flexible rooms in the cyclic high-level schedule. Then, in the *Dynamic* strategy, we decrease the length of the planning periods, allowing us to construct the RMS more frequently. Finally, by implementing the *Agile* planning strategy, we decrease the planning delay, which allows us to construct schedules with more precise information about the waiting lists.

When generating the Refined Master Schedule (RMS) in the two-level planning procedure, we solve the LMSP with the variables β_{ujd}, y_{ud} and μ^{OR}_{jd} fixed. We refer to the LMSP with fixed high-level variables as the Refined Master Scheduling Problem (RMSP).

4. Evaluating the planning framework

In this section, we describe how we evaluate the two-level planning framework in a real-life setting. Then, we present a heuristic solution approach for solving the RMSP, allowing us to speed up the evaluation procedure.

4.1. The evaluation procedure

Before presenting the evaluation procedure, we briefly describe a discrete-event simulation (DES) model, which is an essential part of the procedure. The DES model is developed to represent the operational scheduling of patients from the waiting lists. The entities of the model are the patients, and the state is the number of patients on each waiting list. The state is updated with fixed time increments of one day. In each time increment, new referrals are generated, patients are added to the waiting lists (if the activity delay has passed) and removed according to the activities assigned (the variables x_{jad} in the RMS) using a FIFO scheduling policy. We assume that patient referrals arrive independently of each other, and model the daily arrivals to each specialty as Poisson processes. Upon arrival, we sample the activities that each patient will require, but this information is only used to transfer patients to the waiting lists in a given sequence. We assume that all patients who are scheduled will be served, and if no patients are present to be scheduled, the capacity is lost.

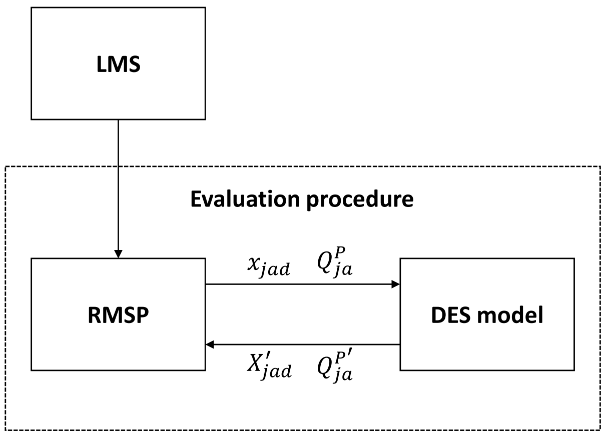


Fig. 4. The procedure for evaluating the planning framework.

The procedure for evaluating the planning framework is illustrated in Fig. 4. The planning framework is evaluated for a given planning horizon and a given number of planning periods, and we have an arbitrary LMS that covers the planning horizon. The LMS does not have to be generated by the LMSP, and it is sufficient to only have the high-level variables available. The high-level variables and a set of initial conditions are fed to the RMSP, generating an RMS for the upcoming planning period. Then, the same initial conditions and the activity type assignments of the RMS are fed to the DES model, which evaluates the performance of the RMS and returns an updated set of initial conditions for the next planning period. The iterations between the RMSP and the DES model continue until all planning periods are simulated and the procedure terminates.

At the time of planning, we must calculate the expected waiting lists for the upcoming planning period in accordance with Eqs. (25), and therefore we require two sets of initial conditions. First, we require the number of patients on each waiting list at the time of planning, Q^P_{ja}. This is an external input when considering the first planning period and is generated by the DES model for the subsequent planning periods. The second set of initial conditions is a predefined schedule of activities for the days covering the maximum activity delay leading up to the time of planning and the planning delay. This set is represented by the X_{jad} parameters and is an external input when considering the first planning period. For later planning periods, it is obtained from the DES model and the previous RMS.

To evaluate the system performance obtained with the LMS, we gather output data from the DES model. This allows us to replicate the objective function value as defined in the LMSP, but we can also extract waiting time distributions for all waiting lists.

4.2. Rolling horizon heuristic

To obtain a precise evaluation of the planning framework, the evaluation procedure must be run multiple times for a given LMS. In each run, the RMSP is solved multiple times, and to obtain an efficient evaluation, we require a method that can find high-quality solutions to the RMSP fast.

To speed up the RMSP, we propose a rolling-horizon heuristic (RHH). Fig. 5 illustrates the RHH for solving the RMSP with a planning horizon of 12 weeks. The procedure iterates through each week of the planning horizon and solves the RMSP once in each iteration. In each iteration, the RMSP considers the remaining weeks of the planning horizon, including the current week, which is indicated in blue. The low-level integer variables for all but the current week are relaxed, as illustrated in green. Before iterating to the next week, we fix all variables for the current week, indicated in gray. The procedure terminates when we have fixed the variables for all weeks in the planning horizon. Note that even if we are able to solve each week in the RHH to optimality, we cannot guarantee that we obtain the optimal solution to the RMSP.

Table 1
The three cases included in the computational study. The number of room-days available in the planning cycle varies to provide 20 instances for each case.

Case	# of rooms		# of room-days			
	R_{OC}	R_{OT}	C^N_{OC}	C^N_{OT}	$\lvert\mathcal{J}\rvert$	$\lvert\mathcal{A}\rvert$
Small	4	3	[14–17]	[4–8]	3	10
Medium	4	4	[17–20]	[7–11]	3	15
Large	8	5	[34–37]	[16–20]	7	22

5. Computational study

In this section, we first perform a technical study of the optimization model. Then, we apply the DES model to validate the optimization model results, before we evaluate the performance of the two-level planning framework, including the flexible, dynamic and agile planning strategies. Throughout this section, we consider a planning horizon of 84 days (12 weeks), a one-week planning cycle, and, except for the agile planning strategy, a planning delay of 28 days. All rooms in the OC and the OT are considered homogeneous, which means that they can accommodate all relevant activity types. Furthermore, we include a set of heterogeneous wards, where each ward can host patients form a subset of the surgery types. We consider three OC activity types for each specialty, including initial (IC), treatment (TC) and follow-up (FU) consultations, and each new referral requires one IC. In contrast to the OC activity types, each surgery type is subordinate to a specific specialty and can only be performed by surgeons that master that specialty. Furthermore, the waiting lists at the time of planning are equal to the target waiting lists, and the waiting lists are measured at the end of each week. The input data are based on real data from the Orthopaedic Clinic at St. Olav's Hospital.

5.1. Technical study of the optimization model

To evaluate the performance of the optimization model, we run it across three cases and 20 instances for each case, presented in Table 1. We study a setting of decentralized surgical departments, and the cases represent the rooms assigned to a specific surgical specialty, in this case the orthopaedic specialty. In the Small case, we have access to four and three rooms in the OC and the OT, respectively. Furthermore, there are three specialties and ten activity types. In the Medium case, there are four rooms available in both units, and there are 15 activity types spread across three specialties. In the Large case, there are eight rooms available in the OC, and five in the OT. Furthermore, there are seven specialties and 22 activity types. The instances within a case are equal, except for the number of room-days available through the planning cycle. To label an instance, we refer to the case, the number of OC-days and the number of OR-days. As an example, the instance L_34_18 refers to the Large case with 34 OC room-days and 18 OR-days. The input data applied in the instances is based on data from the Orthopaedic Clinic.

In the following, we first evaluate the performance of the RHH for solving the RMSP. Then, we introduce a two-level procedure for solving the LMSP, and we evaluate the performance of this procedure.

5.1.1. Performance of the RHH

In the following analysis, we solve the RMSP for the entire planning horizon and evaluate the RHH against solving the full RMSP. For each instance, we run the LMSP for three hours and use the solution obtained as the starting point for solving the RMSP. When solving the full RMSP, we let the model run for one minute, five minutes and one hour. When running the RHH, we terminate each iteration

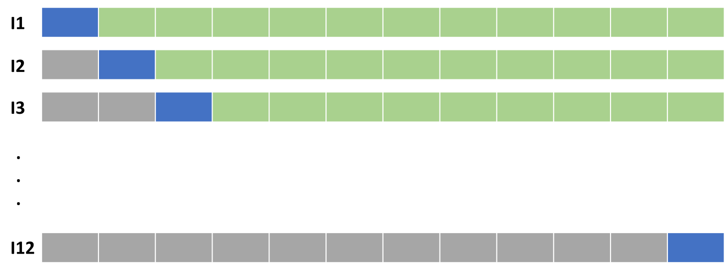


Fig. 5. The RHH for solving the RMSP. We iterate over all weeks in the planning horizon, solving the RMSP for the current (blue) and remaining (green) weeks. In each iteration, the low-level variables are relaxed for all but the current week, and we fix the low-level variables for the current week before going to the next iteration (gray). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

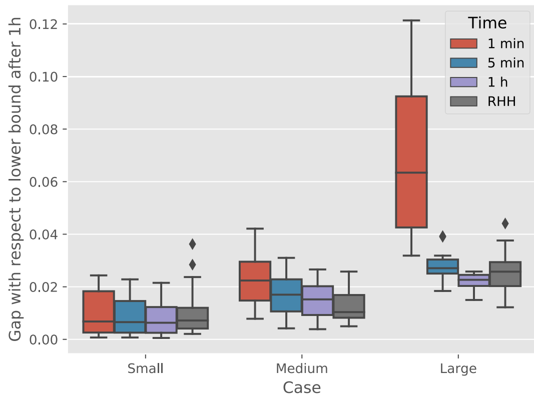


Fig. 6. Comparing the RHH with solving the full RMSP. Each bar represents the distribution of gaps between the primal solutions found when solving the RMSP at the current running time and the lower bound obtained from solving the RMSP for one hour.

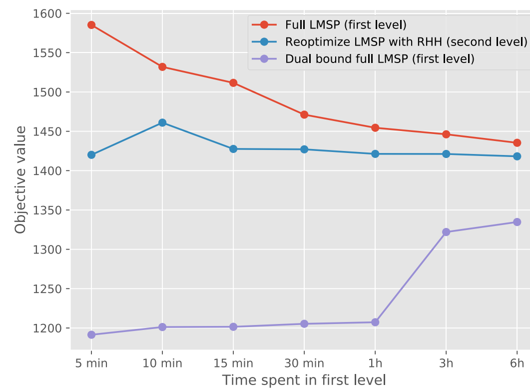


Fig. 7. Objective values obtained from solving the LMSP with the two-level procedure for the L_34_18 instance. The time spent in the first level is increased along the x-axis. Note that the x-axis is non-linear.

after 30 s for the first five iterations and after 10 s for the remaining seven iterations. However, many iterations are solved to optimality before reaching termination. For the Small case instances, the iterations are solved quickly to optimality, and the procedure terminates after seconds. For the Large case instances, some iterations are solved quickly to optimality, and most of the runs terminate within two minutes. In Fig. 6, we display the distribution of gaps across the 20 instances for all three cases and four running times (one min, five min, one hour and RHH). To have a common benchmark for each case and instance, we use the lower bound obtained from solving the full RMSP for one hour, and we calculate the gaps between the lower bound and the primal solution obtained at the current running time. For the Large and Medium cases, the RHH performs well compared to solving the RMSP for one or five minutes. For the Medium case, it even performs well compared to the one hour solution. For the Small case, solving the full RMSP provides narrow gaps fast, leaving little use of the RHH. Based on the findings, we use the RHH when solving the RMSP in all subsequent analyses.

5.1.2. A two-level procedure for solving the LMSP

When solving the LMSP, we exploit the two-level structure of the model and apply a two-level solution procedure. First, we solve the full LMSP. For real-life instances, we cannot solve this problem to optimality, and we terminate the search after a given time. Then, we fix the high-level variables obtained after solving the full LMSP, and we use the RHH to reoptimize the remaining LMSP.

Fig. 7 illustrates, for the L_34_18 instance, the objective values obtained for the LMSP with different times spent in the first level. The red line represents the objective values from solving the full LMSP in the first level, while the blue line is the objective values obtained after reoptimizing the LMSP with the RHH in the second level. When running the RHH, we apply the same time limits as in Section 5.1.1. The purple line is the lower bound obtained in the first level. Not surprisingly, the objective value from solving the full LMSP decreases, while the lower bound increases as we extend the time spent in the first level. However, this monotonous behavior is not reflected after reoptimizing the LMSP in the second level. As a consequence, solutions found early in the search may prove to be better than solutions found later. Furthermore, we seem to obtain good solutions fast, while tightening the LP bound is tedious.

In Fig. 8, we display the distribution of gaps across the 20 instances for all three cases and a set of running times applied when solving the full LMSP. The primal solutions are obtained from reoptimizing the LMSP with the RHH in the second level for each run. To have a common benchmark for each case and instance, we use the lower bound obtained from solving the full LMSP for three hours (Small case) and six hours (Medium and Large cases), and we calculate the gaps between the lower bound and the primal solution obtained at the current running time. In accordance with the results displayed in Fig. 7, we obtain good primal solutions fast, especially for the Small and Medium cases, while for the Large case there is a gain in increasing the running time. For the remaining analysis, we run the LMSP for one hour.

5.2. Validating the optimization model and the objective function

In this section, we validate that LMSs that correspond to good objective values in the optimization model also obtain good objective values when they are simulated in the evaluation procedure. Then, we investigate how well the objective function represents the waiting times and throughput of patients. Finally, we evaluate all 20 Medium case instances obtained from the optimization model results to analyze how the room-day capacities impact on the performance of the system.

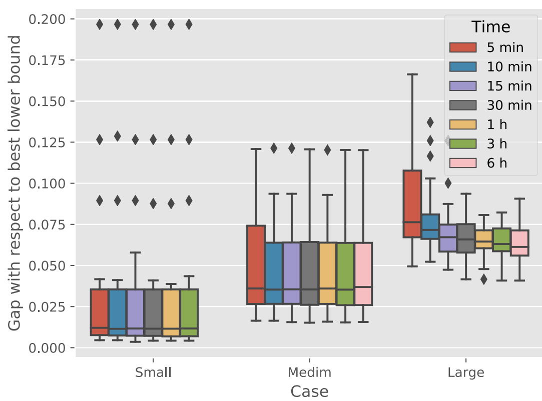


Fig. 8. Comparing the optimality gaps for the LMSP, obtained with the two-level solution procedure. Each box represents a given time by which we allow the first level to run before it is terminated.

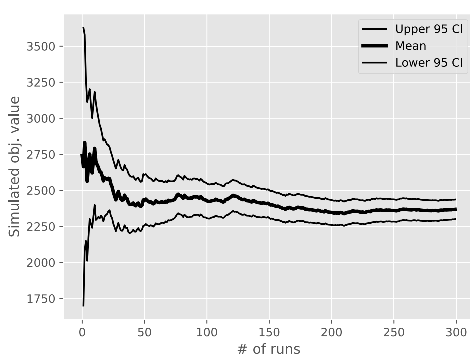


Fig. 9. The accumulated average and 95% confidence interval of the simulated objective value for the M_20_8 instance.

To obtain a precise evaluation of an LMS, the evaluation procedure must be run multiple times. There is a trade-off between the running time required to evaluate an LMS and the precision of the evaluation. By testing the evaluation procedure on multiple instances, we conclude that 200 runs is sufficient. In Fig. 9, we illustrate the accumulated average of the simulated objective value and the 95% confidence interval as we increase the number of simulations for the M_20_8-instance.

To verify the fit between the optimization and simulation model results, we evaluate whether the rank between a set of solutions is consistent in the two models. First, we run the M_20_8 instance with the LMSP and register every 10th solution obtained, with its corresponding objective value. Then, for each solution, we evaluate the corresponding LMS with the evaluation framework and obtain a simulated objective value. In Fig. 10(a), we compare the objective values obtained from optimization and simulation. It is evident that the rank is consistent in the two models, but the values are higher in the simulated results. The reason for this is the stochastic processes introduced in the simulation model, causing the waiting lists to randomly fluctuate. For many waiting lists, this means repeatedly crossing the threshold levels, which shifts the penalty coefficients in the objective function.

Next, we investigate how well the objective function value reflects patient waiting times and throughput obtained from the simulation model. In Fig. 10(b), we display the share of patients that wait for more than 21 days to receive a consultation or surgery (corresponding to the upper threshold) and the number of patients that have received all requested activities. It is clear from the results that the objective function well represents a combination of both measures. Based on these analysis, we argue that the results from the optimization model can be used to predict the real-life performance of the LMS and that the objective function is a good measure for the overall performance of the system.

5.3. Evaluating the planning framework

In this section, we study the Medium case to demonstrate how the proposed planning framework can be used in a real-life setting. First, we run the evaluation procedure for all 20 instances, presenting how this evaluation can be used to provide strategic decision support. Then, we proceed with the five instances with 20 OC room-days, and use these as the base cases for evaluating the performance of the different planning strategies. As our baseline strategy, we optimize with no flexible rooms, planning periods of 28 days, and a planning delay of 28 days. The baseline strategy serves as a basis for comparison, and is marked with a thick black line in Figs. 12 and 14.

Strategic decision support can be obtained from evaluating the planning framework over a set of different instances. As an example, a relevant question can be whether to access one more OC room-day or one additional OR-day to obtain the best performance. In Fig. 11, we display the simulated objective values obtained from the evaluation procedure, across all 20 instances for the Medium case. In general, the objective values decrease if we increase the room-day capacities, as expected. However, given an instance, it is not indifferent whether we should access one more OC room-day or one additional OR-day to obtain the best performance. In this situation, such an analysis can prove to be valuable.

To evaluate the flexible strategy, we assign one or two flexible rooms to each of the units when constructing the LMS. This lets us postpone the assignment of specialties to these rooms to the points in time when we construct the RMS. Flexibility allows for more freedom when adjusting the resource assignments to the current waiting lists. The results are displayed in Fig. 12(a), and we see that adding flexibility has a value. However, there is a diminishing marginal value of adding flexibility, and adding a limited number of flexible rooms is sufficient. Note also that the value of flexibility is not constant across the different resource instances.

In the dynamic strategy, we decrease the planning period, allowing us to increase the frequency with which we construct the RMS. Here, we evaluate the effects of decreasing the planning periods to 14 and seven days. The results of dynamic planning are illustrated in Fig. 12(b). We see that decreasing the planning period yields better results; however, this is not the case when going from 14 to seven days. The reason for this is that our modeling approach tends to be myopic for short planning periods.

The agile planning strategy refers to decreasing the planning delay. A more agile system will have more accurate information when planning the upcoming planning period. To evaluate the agile planning strategy, we apply a 14- and zero-day planning delay. The results of an agile planning strategy are shown in Fig. 12(c). We see that reducing the planning delay yields increased value across all instances. Having a zero-day planning delay is unrealistic, but we choose to include it to show the potential of agility.

The strategies described above can be combined to obtain additive, or even amplified effects. In Fig. 13, we display the average objective function values across the five base instances for different values of dynamics, agility and flexibility. The red, yellow and green surfaces represent zero, one and two flexible rooms in each unit, respectively. The value of decreasing the planning delay (imposing agility) is more pronounced for a highly dynamic strategy than for a completely static strategy (84-day planning period). Likewise, the value of increasing dynamics is higher for an agile than for a non-agile planning strategy. These results indicate amplified effects of dynamics and agility. Furthermore, the value of going from zero to one flexible room in each unit adds value across all combinations of planning delays and planning periods. However, increasing from one to two flexible rooms does not have any additional effects in any of the combinations. In Fig. 14, we illustrate a combined planning strategy with one flexible room in both units, two-week planning periods and a two-week planning delay. It is evident that combining the planning strategies yields an additive value for the system. In fact, the combined strategy performs better with 9 OR-days than the base strategy with 11 OR-days.

6. Conclusion

In this paper, we develop a framework for integrated resource planning in surgical clinics. The framework considers the integrated planning of the OC and the OT, and is composed of an optimization model for solving the LMSP, and a two-level planning procedure to handle stochastic waiting lists. Furthermore, we develop a simulation-based evaluation procedure and use this to evaluate three planning strategies that can be adopted in the planning framework.

Compared to a baseline strategy, the adoption of any of the three planning strategies improves the outcomes of the system. By implementing a combination of agility and dynamics, we obtain additive effects, and the strategies amplify each other. Furthermore, imposing a little flexibility improves the outcomes of the system in all combinations of agility and dynamics.

While the mathematical model in this paper is deterministic, the planning framework is designed to handle fluctuating waiting lists by frequently updating the RMS. When simulating, we only introduce stochastic arrivals and patient requirements, while keeping the activity durations and patient LOS deterministic. By introducing more sources of uncertainty we believe that the waiting lists will fluctuate more, strengthening the value of flexibility, dynamics and agility.

The planning strategies proposed are based on flexible and dynamic planning procedures and challenge conventional ways of planning in hospitals. We believe that the agile planning strategy is the most challenging strategy. Our impression is that both flexible and dynamic strategies can be adopted if the staff is noticed about the adjustments some time before they are executed. However, the value of such adjustments has less impact in the case of an excessive planning delay. Therefore, we propose that hospital clinics establish procedures that enable agility.

There is clearly a trade-off between complexity and efficiency when it comes to planning. A more sophisticated procedure can yield higher performance, but the adoption threshold may be too high. Studying this trade-off can prove valuable for hospitals that want to improve their planning procedures and is a topic for future research. Adopting a short planning delay has practical issues. Interestingly, the planning structure with a delay between planning and execution is well suited for stochastic programming. Can stochastic programming be used to obtain efficient schedules with longer planning delays? Furthermore, an interesting avenue for future research is to test the framework under different conditions to identify how the value of the planning strategies depends on the system characteristics. Finally, the development of efficient solution procedures for the LMSP is relevant for future research. The model formulation is generic and can be applied to study larger cases where the rooms are not yet assigned to surgical specialties, which is the case for a centralized surgical clinic. However, our analyses show that scalability is an issue when striving for optimal solutions.

CRediT authorship contribution statement

Thomas Reiten Bovim: Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Conceptualization. **Anders N. Gullhav:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Henrik Andersson:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Atle Riise:** Writing – original draft, Supervision, Methodology, Conceptualization.

Acknowledgments

We would like to thank the management in the Orthopaedic Department at St. Olav's Hospital, and in particular Head of Department, Vigleik Jessen, MD for good collaboration throughout the project. We are grateful for the comments and suggestions made by two anonymous reviewers, which helped to improve the paper.

Appendix A. The notation related to the LMSP

The notation used in the LMSP is presented in Tables A.2 to A.5.

A.1. The mathematical model

$$min \quad \frac{1}{|\mathcal{D}^Q|} \sum_{j\in\mathcal{J}} \sum_{a\in\mathcal{A}_j^J} \sum_{d\in\mathcal{D}^Q} \sum_{k\in\mathcal{K}} \bar{C}_{jak} \bar{q}_{jadk} \tag{A.1}$$

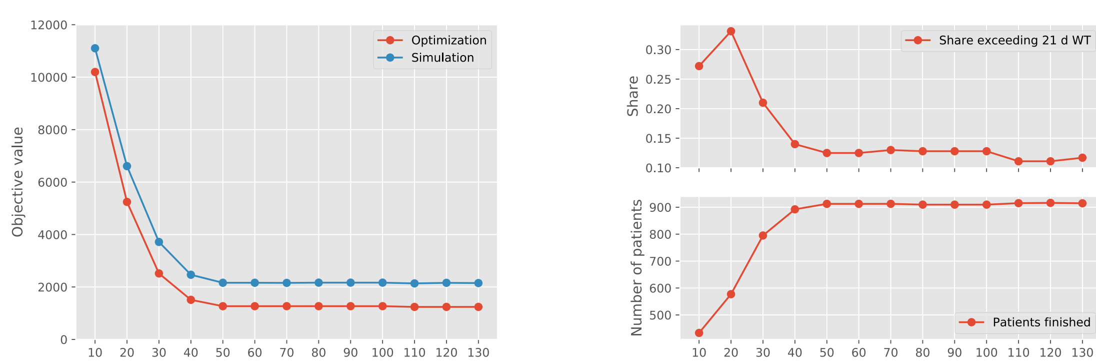
$$\sum_{j\in\mathcal{J}} \beta_{ujd} + y_{ud} \le R_u \qquad u \in \mathcal{U}, d \in \mathcal{D}^C \tag{A.2}$$

$$\sum_{d\in\mathcal{D}^C} y_{ud} = B_u^F \qquad u \in \mathcal{U} \tag{A.3}$$

$$\sum_{j\in\mathcal{J}} \sum_{d\in\mathcal{D}^C} \beta_{ujd} + \sum_{d\in\mathcal{D}^C} y_{ud} \le C_u^N \qquad u \in \mathcal{U} \tag{A.4}$$

$$\mu_{jd}^{OR} \le \sum_{p\in\mathcal{P}_j^C} C_{pd} - \beta_{ujd} \qquad u = OT, j \in \mathcal{J}, d \in \mathcal{D}^C \tag{A.5}$$

$$\sum_{j\in\mathcal{J}} \mu_{jd}^{OR} = \sum_{p\in\mathcal{P}^C} C_{pd} - \sum_{j\in\mathcal{J}} \beta_{ujd} \qquad u = OT, d \in \mathcal{D}^C \tag{A.6}$$



(a) Comparing the objective values from the optimization model to the simulated objective values. (b) Top: Share of consultations exceeding 21 days waiting time. Bottom: Number of patients that are finished.

Fig. 10. Validating the optimization model and the objective function.

Table A.2
Sets.

Symbol	Description	
$\mathcal{D}$	Days covering the maximum activity delay and the planning horizon	$d \in \mathcal{D}$
$\mathcal{D}^T$	Days in the planning horizon	$d \in \mathcal{D}^T \subseteq \mathcal{D}$
$\mathcal{D}^C$	Days in a planning cycle	$d \in \mathcal{D}^C$
$\mathcal{D}^P$	Days in the planning delay	$d \in \mathcal{D}^P$
$\mathcal{D}^T_{d'}$	Days in the planning horizon that correspond to cycle day d'	$d \in \mathcal{D}^T \mid d \mod \lvert D^C \rvert = d'$
$\mathcal{D}^Q$	Days when we measure the waiting lists	$d \in \mathcal{D}^Q \subseteq \mathcal{D}^T$
$\mathcal{D}^{LOS}_{jad}$	Days that a patient who stays in a ward on day d can have had a surgery of specialty j and surgery type a	$d' \in \mathcal{D}^{LOS}_{jad}$
$\mathcal{U}$	Units	$u \in \mathcal{U}$
$\mathcal{J}$	Surgical specialties	$j \in \mathcal{J}$
$\mathcal{P}$	Surgeon types	$p \in \mathcal{P}$
$\mathcal{W}$	Wards	$w \in \mathcal{W}$
$\mathcal{B}$	OR activity blocks	$b \in \mathcal{B}$
$\mathcal{A}$	Activity types	$a \in \mathcal{A}$
$\mathcal{A}^{OC}$	OC activity types	$a \in \mathcal{A}^{OC} \subseteq \mathcal{A}$
$\mathcal{A}^{OT}$	Surgery activity types	$a \in \mathcal{A}^{OT} \subseteq \mathcal{A}$
$\mathcal{K}$	Waiting list intervals	$k \in \mathcal{K}$
$\mathcal{P}^C$	Consultant types	$p \in \mathcal{P}^C \subseteq \mathcal{P}$
$\mathcal{P}^C_j$	Consultant types that can cover specialty j	$p \in \mathcal{P}^C_j \subseteq \mathcal{P}$
$\mathcal{P}^R_j$	Resident types that can cover specialty j	$p \in \mathcal{P}^R_j \subseteq \mathcal{P}$
$\mathcal{W}^A_a$	Wards that can house patients who received activity type a	$w \in \mathcal{W}^A_a \subseteq \mathcal{W}$
$\mathcal{B}^J_j$	OR activity blocks available for specialty j	$b \in \mathcal{B}^J_j \subseteq \mathcal{B}$
$\mathcal{A}^J_j$	Activity types that can be handled by specialty j	$a \in \mathcal{A}^J_j \subseteq \mathcal{A}$
$\mathcal{A}^{OT}_j$	Surgery activity types that can be handled by specialty j	$a \in \mathcal{A}^{OT}_j \subseteq \mathcal{A}^J_j$
$\mathcal{A}^W_w$	Surgery activity types that can rest in ward w following surgery	$a \in \mathcal{A}^W_w \subseteq \mathcal{A}^{OT}$

Table A.3
Parameters.

Symbol	Description
R_u	Number of rooms available in unit u
V_{ujd}	Number of rooms that can be accessed in unit u by specialty j on cycle day d
C^N_u	Number of room-days that can be accessed in unit u during the planning cycle
B^F_u	Number of room-days that must be assigned as flexible in unit u through the planning cycle
T^{OC}	Time available in an OC room-day
C_{pd}	Number of surgeons available of surgeon type p on cycle day d
C^{MAX}_p	Maximum number of days available for surgeon type p during the planning horizon
N^B_b	Number of surgeons that must be present to assign OR activity block b
A_{wd}	Number of staffed beds available in ward w on day d in the planning cycle
X_{jad}	Number of activities of type a and specialty j (expected to be) performed on day d, before the planning horizon
L_{ja}	Expected external arrival rate of activity type a and specialty j
$F_{jaa'}$	Fraction of activity of type a that yields a downstream demand for activity of type a' for specialty j
D^{OC}_{ja}	Duration of OC activity type a, specialty j
A^B_{bja}	Number of patients from specialty j and activity type a that are assigned to OR activity block b
D^P	Number of days in the planning delay
D^A_{ja}	Number of days in the activity delay after activity type a, specialty j
Q^0_{ja}	Number of patients on the waiting lists for specialty j and activity type a when entering the planning horizon
Q_{jak}	Maximum number of patients that can be assigned to the waiting list of specialty j, activity type a and waiting list interval k
W_{jak}	Threshold waiting time of specialty j, activity type a and waiting list interval k
$\overline{\lambda}_{ja}$	Average arrival rate to waiting list of specialty j and activity type a
$\overline{C}_{jak}$	Penalty coefficient associated with the waiting list of specialty j, activity type a and waiting list interval k

Table A.4
The high-level variables.

Symbol	Description
β_{ujd}	Number of rooms assigned to unit u and specialty j on cycle day d
y_{ud}	Number of rooms in unit u assigned as flexible on cycle day d
μ^{OR}_{jd}	Maximum number of ORs that can be assigned as flexible for specialty j on cycle day d

Table A.5
The low-level variables.

Symbol	Description
λ_{ujd}	Number of rooms in unit u used by specialty j on day d
y_{ujd}	Number of flexible rooms in unit u assigned for specialty j on day d
g_{pjd}	Number of surgeons from surgeon type p assigned to specialty j on day d
x^{OR}_{bd}	Number of OR blocks of type b assigned to day d
x_{jad}	Number of activities of type a assigned to specialty j on day d
u_{jawd}	Number of beds occupied in ward w on day d, by patients of specialty j who received surgery type a
q_{jad}	Number of patients on the waiting list of specialty j and activity type a on day d
$\overline{q}_{jadk}$	Number of patients on the waiting list of specialty j and activity type a on day d, within interval k

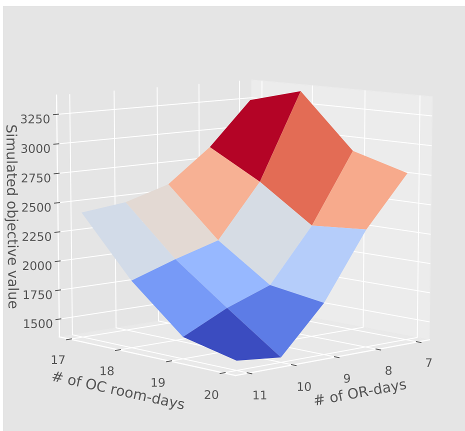


Fig. 11. The simulated objective values across all instances for the Medium case.

Table B.6
The main sets.

Set	Symbol	# of elements		
		Small	Medium	Large
Days in planning horizon	$\mathcal{D}^T$	84	84	84
Days in planning cycle	$\mathcal{D}^C$	7	7	7
Days when we measure the waiting lists	$\mathcal{D}^Q$	12	12	12
Surgical specialties	$\mathcal{J}$	3	3	7
OC activity types	$\mathcal{A}^{OC}$	3	3	3
Surgery activity types	$\mathcal{A}^{OT}$	7	12	19
Wards	$\mathcal{W}$	3	3	4
Surgeon types	$\mathcal{P}$	6	6	14
Waiting list intervals	$\mathcal{K}$	3	3	3

$$\sum_{j'\in\mathcal{J}|j'\neq j} \mu^{OR}_{j'd} \geq y_{ud} \qquad u = OT, j \in \mathcal{J}, d \in \mathcal{D}^C \qquad \text{(A.7)}$$

$$\lambda_{ujd} \leq \beta_{ujd'} + y_{ujd} \qquad u \in \mathcal{U}, j \in \mathcal{J}, d' \in \mathcal{D}^C, d \in \mathcal{D}^T_{d'} \qquad \text{(A.8)}$$

$$\sum_{j\in\mathcal{J}} y_{ujd} \leq y_{ud'} \qquad u \in \mathcal{U}, d' \in \mathcal{D}^C, d \in \mathcal{D}^T_{d'} \qquad \text{(A.9)}$$

$$\lambda_{ujd} + \sum_{b\in\mathcal{B}^J_j} N^B_b x^{OR}_{bd} \leq \sum_{p\in\mathcal{P}^C_j\cup\mathcal{P}^R_j} g_{pjd} \qquad u = OC, j \in \mathcal{J}, d \in \mathcal{D}^T \qquad \text{(A.10)}$$

$$\sum_{b\in\mathcal{B}^J_j} x^{OR}_{bd} \leq \sum_{p\in\mathcal{P}^C_j} g_{pjd} \qquad j \in \mathcal{J}, d \in \mathcal{D}^T \qquad \text{(A.11)}$$

$$\sum_{j\in\mathcal{J}} g_{pjd} \leq C_{pd'} \qquad p \in \mathcal{P}, d' \in \mathcal{D}^C, d \in \mathcal{D}^T_{d'} \qquad \text{(A.12)}$$

$$\sum_{j\in\mathcal{J}} \sum_{d\in\mathcal{D}^T} g_{pjd} \leq C^{MAX}_p \qquad p \in \mathcal{P} \qquad \text{(A.13)}$$

$$\sum_{b\in\mathcal{B}^J_j} x^{OR}_{bd} = \lambda_{ujd} \qquad u = OT, j \in \mathcal{J}, d \in \mathcal{D}^T \qquad \text{(A.14)}$$

$$x_{jad} = X_{jad} \qquad j \in \mathcal{J}, a \in \mathcal{A}, d \in \mathcal{D}|d < 1 \qquad \text{(A.15)}$$

$$x_{jad} = \sum_{b\in\mathcal{B}^J_j} A^B_{bja} x^{OR}_{bd} \qquad j \in \mathcal{J}, a \in \mathcal{A}^{OT}_j, d \in \mathcal{D}^T \qquad \text{(A.16)}$$

$$\sum_{a\in\mathcal{A}^{OC}} D^{OC}_{ja} x_{jad} \leq T^{OC} \lambda_{ujd} \qquad u = OC, j \in \mathcal{J}, d \in \mathcal{D}^T \qquad \text{(A.17)}$$

$$\sum_{d'\in\mathcal{D}^{LOS}_{jad}} x_{jad'} = \sum_{w\in\mathcal{W}^A_a} u_{jawd} \qquad j \in \mathcal{J}, a \in \mathcal{A}^{OT} \cap \mathcal{A}^J_j, d \in \mathcal{D}^T \qquad \text{(A.18)}$$

$$\sum_{j\in\mathcal{J}} \sum_{a\in\mathcal{A}^W_w\cap\mathcal{A}^J_j} u_{jawd} \leq A_{wd'} \qquad w \in \mathcal{W}, d' \in \mathcal{D}^C, d \in \mathcal{D}^T_{d'} \qquad \text{(A.19)}$$

$$q_{jad} \geq Q^0_{ja} - x_{jad} + L_{ja} + \sum_{a'\in\mathcal{A}^J_j} F_{ja'a} x_{ja',(d-D^A_{ja})} \qquad j \in \mathcal{J}, a \in \mathcal{A}^J_j, d = 1 \qquad \text{(A.20)}$$

$$q_{jad} \geq q_{ja,(d-1)} - x_{jad} + L_{ja} + \sum_{a'\in\mathcal{A}^J_j} F_{ja'a} x_{ja',(d-D^A_{ja})} \qquad j \in \mathcal{J}, a \in \mathcal{A}^J_j, d \in \mathcal{D}^T|d > 1 \qquad \text{(A.21)}$$

$$Q_{jak} = W_{jak} \cdot \overline{\lambda}_{ja} \qquad j \in \mathcal{J}, a \in \mathcal{A}^J_j, k \in \mathcal{K} \qquad \text{(A.22)}$$

$$q_{jad} = \sum_{k\in\mathcal{K}} \overline{q}_{jadk} \qquad j \in \mathcal{J}, a \in \mathcal{A}^J_j, d \in \mathcal{D}^Q \qquad \text{(A.23)}$$

$$\overline{q}_{jadk} \leq Q_{jak} \qquad j \in \mathcal{J}, a \in \mathcal{A}^J_j, d \in \mathcal{D}^Q, k \in \mathcal{K} \qquad \text{(A.24)}$$

Parameter V_{ujd} represents the maximum number of rooms that can be assigned to specialty j in unit u on cycle day d, and it is an upper bound for the variables β_{ujd}, λ_{ujd} and x^{OR}_{bd}. For the OC, it is the minimum of the number of surgeons that can cover specialty j available on cycle day d, and the number of OC rooms available, R_{OC}. For the OT, it is the minimum of the number of consultants that can cover specialty j available on cycle day d, and the number of ORs available, R_{OT}.

$$V_{ujd} = \min\left\{ \sum_{p\in\mathcal{P}^C_j\cup\mathcal{P}^R_j} C_{pd}, R_u \right\} \qquad u = OC, j \in \mathcal{J}, d \in \mathcal{D}^C \qquad \text{(A.25)}$$

$$V_{ujd} = \min\left\{ \sum_{p\in\mathcal{P}^C_j} C_{pd}, R_u \right\} \qquad u = OT, j \in \mathcal{J}, d \in \mathcal{D}^C \qquad \text{(A.26)}$$

$$\beta_{ujd} \in \{0, 1, \ldots, V_{ujd}\} \qquad u \in \mathcal{U}, j \in \mathcal{J}, d \in \mathcal{D}^C \qquad \text{(A.27)}$$

$$y_{ud} \in \{0, 1, 2, \ldots\} \qquad u \in \mathcal{U}, d \in \mathcal{D}^C \qquad \text{(A.28)}$$

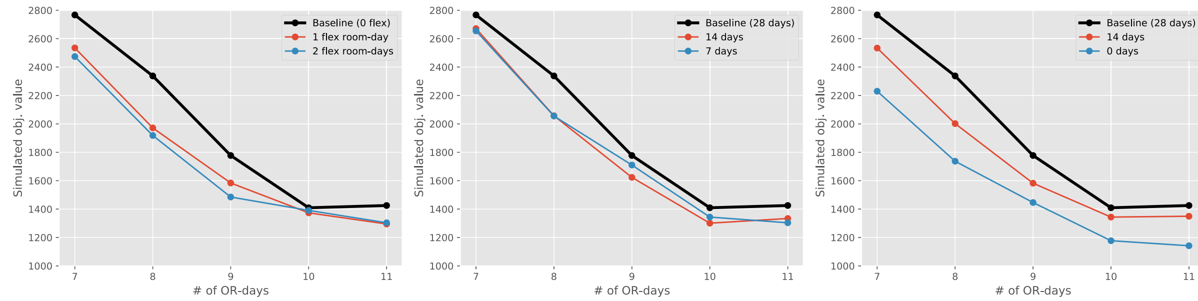
$$\mu^{OR}_{jd} \geq 0 \qquad j \in \mathcal{J}, d \in \mathcal{D}^C \qquad \text{(A.29)}$$

$$\lambda_{ujd} \in \{0, 1, \ldots, V_{ujd'}\} \qquad u \in \mathcal{U}, j \in \mathcal{J}, d' \in \mathcal{D}^C, d \in \mathcal{D}^T_{d'} \qquad \text{(A.30)}$$

$$y_{ujd} \in \{0, 1, 2, \ldots\} \qquad u \in \mathcal{U}, j \in \mathcal{J}, d \in \mathcal{D}^T \qquad \text{(A.31)}$$

$$g_{pjd} \in \{0, 1, \ldots, C_{pd}\} \qquad j \in \mathcal{J}, p \in \mathcal{P}^C_j \cup \mathcal{P}^R_j, d \in \mathcal{D}^T \qquad \text{(A.32)}$$

$$x^{OR}_{bd} \in \{0, 1, \ldots, V_{OT,jd'}\} \qquad j \in \mathcal{J}, b \in \mathcal{B}_j, d' \in \mathcal{D}^C, d \in \mathcal{D}^T_{d'} \qquad \text{(A.33)}$$



(a) Flexible strategy: The value of assigning flexible rooms in the LMS.

(b) Dynamic strategy: The value of decreasing the planning period

(c) Agile strategy: The value of decreasing the planning delay.

Fig. 12. Evaluating the three different planning strategies.

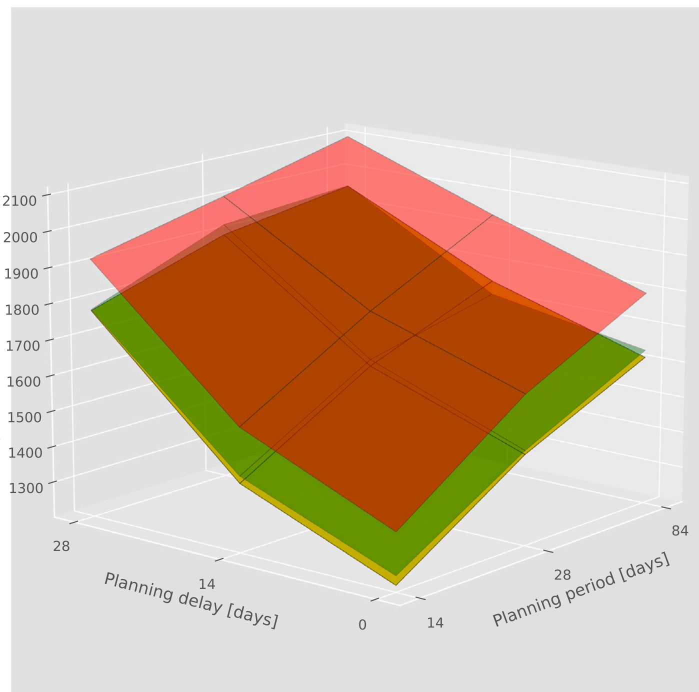


Fig. 13. Combined affects of dynamics and agility for different number of flexible rooms. The values on the z-axis are the average objective values from simulating the five base instances over a set of different planning strategies. The red, yellow and green surfaces represent zero, one and two flexible rooms in each unit, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table B.7
The availability of surgeons on weekdays, and the maximum number of days available for clinical work in a planning period. C refers to consultants, while R is residents.

Surgeon type	Mon	Tue	Wed	Thu	Fri	Days in planning period		
						Small	Medium	Large
Arthroscopy C	2	2	1	3	2	30	30	20
Arthroscopy R	2	1	2	1	2	15	15	15
Hand C	2	1	3	1	1	30	30	24
Hand R	1	2	1	2	1	16	16	16
Plastic C	2	2	2	1	3		35	30
Plastic R	3	1	2	2	2		34	34
Arthroplasty C	2	2	2	1	0			24
Arthroplasty R	2	2	1	1	0			24
Reconstructive C	2	2	0	1	1	20		20
Reconstructive R	0	1	1	1	0	16		26
Back C	1	2	1	2	1			24
Back R	2	1	2	1	1			16
Tumor C	1	0	1	0	1			15
Tumor R	0	1	0	1	1			12

Table B.8
The number of beds available.

Ward	Small							Medium							Large						
	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
Trauma	4	4	4	4	4	2	2	4	4	4	4	4	2	2	4	4	4	4	4	2	2
Reconstructive	5	5	5	5	5	3	3	4	4	4	4	4	3	3	5	5	5	5	5	3	3
Elective	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
FT															16	16	16	16	16	0	0

Table B.9
The rooms.

Location	Availability [min]	Number of rooms			Number of room-days		
		Small	Medium	Large	Small	Medium	Large
Outpatient clinic	240	4	4	8	14–17	17–20	34-37
Operating theater	480	3	4	5	4–8	7–11	16-20

Table B.10
OC activity types.

Activity type	Specialty	Duration [min]	Q_{ja1}	Q_{ja2}	Q_{ja3}	$\overline{C}_{ja1}$	$\overline{C}_{ja2}$	$\overline{C}_{ja3}$
IC	Arthroscopy	30	34.0	17.0	∞	5	15	100
TC	Arthroscopy	30	1.7	0.9	∞	2	5	50
FU	Arthroscopy	30	26.6	13.3	∞	0	1	30
IC	Hand	30	38.0	19.0	∞	5	15	100
TC	Hand	30	20.1	10.1	∞	2	5	50
FU	Hand	30	78.8	39.4	∞	0	1	30
IC	Plastic	30	58.0	29.0	∞	5	15	100
TC	Plastic	30	11.6	5.8	∞	2	5	50
FU	Plastic	30	90.1	45.0	∞	0	1	30
IC	Arthroplasty	30	38.0	19.0	∞	5	15	100
TC	Arthroplasty	30	1.9	1.0	∞	2	5	50
FU	Arthroplasty	30	29.3	14.6	∞	0	1	30
IC	Reconstructive	30	36.0	18.0	∞	5	15	100
TC	Reconstructive	30	2.2	1.0	∞	2	5	50
FU	Reconstructive	30	41.0	20.6	∞	0	1	30
IC	Back	30	10.0	5.0	∞	5	15	100
TC	Back	30	0.5	0.3	∞	2	5	50
FU	Back	30	13.8	6.9	∞	0	1	30
IC	Tumor	30	10.0	5.0	∞	5	15	100
TC	Tumor	30	5.3	2.7	∞	2	5	50
FU	Tumor	30	40.2	20.1	∞	0	1	30

Table B.11
Surgery types.

Surg. type	Specialty	Dur. [min]	# surgeons	Ward	LOS [days]	Q_{ja1}	Q_{ja2}	Q_{ja3}	$\overline{C}_{ja1}$	$\overline{C}_{ja2}$	$\overline{C}_{ja3}$
Arthro. (agg.)	Arthroscopy	174	2	El.	2	5.1	7.7	∞	2	5	50
ACL	Arthroscopy	173	1	El.	2	4.1	6.1	∞	2	5	50
Meniscus	Arthroscopy	103	2	–	0	3.7	5.6	∞	2	5	50
Patellae	Arthroscopy	176	2	El.	1	3.4	5.1	∞	2	5	50
Hand (agg.)	Hand	107	2	–	0	11.4	17.1	∞	2	5	50
CTS	Hand	54	2	Tr.	1	3.8	5.7	∞	2	5	50
Plastic (agg.)	Plastic	108	2	Tr., Recon.	2	17.4	26.1	∞	2	5	50
Carsinoma	Plastic	52	1	Recon.	1	5.8	8.7	∞	2	5	50
BCC	Plastic	59	2	Tr., Recon.	1	2.9	4.4	∞	2	5	50
Mal. mel.	Plastic	85	1	–	0	8.7	13.1	∞	2	5	50
Cancer m.	Plastic	146	1	Recon.	1	5.8	8.7	∞	2	5	50
SCC	Plastic	65	2	Recon., El.	1	2.9	4.4	∞	2	5	50
Hip (primary)	Arthroplasty	110	2	FT	4	14.1	21.1	∞	2	5	50
Hip (revision)	Arthroplasty	152	2	FT	4	3.4	5.3	∞	2	5	50
Knee (primary)	Arthroplasty	122	2	FT	4	8.0	12.0	∞	2	5	50
Knee (revision)	Arthroplasty	165	2	FT	4	1.9	2.9	∞	2	5	50
Recon. (agg.)	Reconstructive	145	2	Recon.	2	5.8	8.6	∞	2	5	50
Back (agg.)	Back	309	2	El.	6	1.8	2.7	∞	2	5	50
Tumor (agg.)	Tumor	93	1	Recon.	1	1.4	2.1	∞	2	5	50

$$x_{jad} \in \{0, 1, \dots\} \qquad j \in \mathcal{J}, a \in \mathcal{A}_j^J, d \in \mathcal{D} \tag{A.34}$$

$$u_{jawd} \in \{0, 1, 2, \dots\} \qquad j \in \mathcal{J}, a \in \mathcal{A}^{OT} \cap \mathcal{A}_j^J, w \in \mathcal{W}, d \in \mathcal{D}^T \tag{A.35}$$

$$q_{jad} \geq 0 \qquad j \in \mathcal{J}, a \in \mathcal{A}_j^J, d \in \mathcal{D}^T \tag{A.36}$$

$$\overline{q}_{jadk} \geq 0 \qquad j \in \mathcal{J}, a \in \mathcal{A}_j^J, d \in \mathcal{D}^Q, k \in \mathcal{K} \tag{A.37}$$

Appendix B. Input data for the LMSP

In Tables B.6 to B.13, we provide the data applied to define the Small, Medium and Large cases in the LMSP.

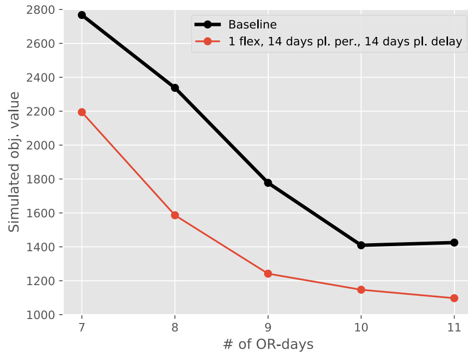


Fig. 14. The value added by combining the planning strategies. Here, we compare the baseline strategy to a strategy with one flexible room in both units, planning periods of two weeks, and a two-week planning delay.

Table B.12
The flow of patients at the OC. A share of 2 in the final column means that we expect two FUs after a TC.

Specialty	Expected # of new IC per day	Share to TC after IC	Share to FU after TC
Arthroscopy	2.43	0.05	1
Hand	2.71	0.53	2
Plastic	4.14	0.20	1
Arthroplasty	2.71	0.05	1
Reconstructive	2.57	0.06	2
Back	1.43	0.05	2
Tumor	1.43	0.53	2

Table B.13
The flow of patients at the operating theater.

Surgery category	Share to surgery after IC	Share to FU after surgery
Arthroscopy (aggregated)	0.15	1
ACL	0.12	1
Meniscus	0.11	1
Patellae	0.10	1
Hand (aggregated)	0.30	1
CTS	0.10	1
Plastic (aggregated)	0.30	1
Carsinoma	0.10	2
BCC	0.05	1
Malignant melanoma	0.15	1
Cancer mammae	0.10	2
SCC	0.05	2
Hip (primary)	0.37	1
Hip (revision)	0.09	1
Knee (primary)	0.21	1
Knee (revision)	0.05	1
Reconstructive (aggregated)	0.32	2
Back (aggregated)	0.18	2
Tumor (aggregated)	0.14	2

References

Agnetis, A., Coppi, A., Corsini, M., Dellino, G., Meloni, C., & Pranzo, M. (2012). Long term evaluation of operating theater planning policies. *Operations Research for Health Care*, *1*(4), 95–104.

Agnetis, A., Coppi, A., Corsini, M., Dellino, G., Meloni, C., & Pranzo, M. (2014). A decomposition approach for the combined master surgical schedule and surgical case assignment problems. *Health Care Management Science*, *17*, 49–59.

Banditori, C., Cappanera, P., & Visintin, F. (2013). A combined optimization–simulation approach to the master surgical scheduling problem. *IMA Journal of Management Mathematics*, *24*(2), 155–187.

Blake, J. T., & Carter, M. W. (1997). Surgical process scheduling: a structured review. *Journal of the Society for Health Systems*, *5*(3), 17–30.

Bovim, T. R., Abdullahu, A., Andersson, H., & Gullhav, A. N. (2022). Integrated master surgery and outpatient clinic scheduling. *Operations Research for Health Care*, *35*, Article 100358.

Cappanera, P., Visintin, F., & Banditori, C. (2014). Comparing resource balancing criteria in master surgical scheduling: A combined optimisation-simulation approach. *International Journal of Production Economics*, *158*, 179–196.

Cardoen, B., Demeulemeester, E., & Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, *201*(3), 921–932.

European Commission (2023). The impact of demographic change in a changing environment.

Fügener, A. (2015). An integrated strategic and tactical master surgery scheduling approach with stochastic resource demand. *Journal of Business Logistics*, *36*(4), 374–387.

Fügener, A., Hans, E. W., Kolisch, R., Kortbeek, N., & Vanberkel, P. T. (2014). Master surgery scheduling with consideration of multiple downstream units. *European Journal of Operational Research*, *239*(1), 227–236.

Hulshof, P. J. H., Kortbeek, N., Boucherie, R. J., Hans, E. W., & Bakker, P. J. M. (2012). Taxonomic classification of planning decisions in health care: a structured review of the state of the art in OR/MS. *Health Systems*, *1*(2), 129–175.

Laan, C., van de Vrugt, M., Olsman, J., & Boucherie, R. J. (2018). Static and dynamic appointment scheduling to improve patient access time. *Health Systems*, *7*(2), 148–159.

Makboul, S., Kharraja, S., Abbassi, A., & Alaoui, A. E. H. (2022). A two-stage robust optimization approach for the master surgical schedule problem under uncertainty considering downstream resources. *Health Care Management Science*, *25*(1), 63–88.

Mazloumian, M., Baki, M. F., & Ahmadi, M. (2022). A robust multiobjective integrated master surgery schedule and surgical case assignment model at a publicly funded hospital. *Computers & Industrial Engineering*, *163*, Article 107826.

Moosavi, A., & Ebrahimnejad, S. (2020). Robust operating room planning considering upstream and downstream units: A new two-stage heuristic algorithm. *Computers & Industrial Engineering*, *143*, Article 106387.

OECD Publishing (2022). Health at a glance: Europe 2022: State of health in the EU cycle. http://dx.doi.org/10.1787/507433b0-en.

Oliveira, M., Visintin, F., Santos, D., & Marques, I. (2021). Flexible master surgery scheduling: combining optimization and simulation in a rolling horizon approach. *Flexible Services and Manufacturing Journal*.

Santos, D., & Marques, I. (2022). Designing master surgery schedules with downstream unit integration via stochastic programming. *European Journal of Operational Research*, *299*(3), 834–852.

Schneider, A. J. T., van Essen, J. T., Carlier, M., & Hans, E. W. (2020). Scheduling surgery groups considering multiple downstream resources. *European Journal of Operational Research*, *282*(2), 741–752.

Spratt, B., & Kozan, E. (2016). Waiting list management through master surgical schedules: A case study. *Operations Research for Health Care*, *10*, 49–64.

Vanberkel, P. T., Boucherie, R. J., Hans, E. W., Hurink, J. L., van Lent, W. A. M., & van Harten, W. H. (2011). An exact approach for relating recovering surgical patient workload to the master surgical schedule. *Journal of the Operational Research Society*, *62*(10), 1851–1860.