Omega 129 (2024) 103154

Contents lists available at ScienceDirect

Omega

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

Optimizing physician schedules with resilient break assignments☆

Sebastian Kraul [a,*], Melanie Erhard [b], Jens O. Brunner [b,c,d]

[a] Department of Operations Analytics, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
[b] Health Care Operations/Health Information Management, Faculty of Business and Economics, Faculty of Medicine, University of Augsburg, Augsburg, Germany
[c] Department of Technology, Management, and Economics, Technical University of Denmark, Lyngby, Denmark
[d] Center for Excellence in Healthcare Operations Planning, Next Generation Technology, Region Zealand, Slagelse, Denmark

ARTICLE INFO

Keywords:
OR in health services
Physicians scheduling
Rostering
Branch-and-price
Flexible break assignment
Survival regression

ABSTRACT

This article presents a novel model for building biweekly rosters for physicians according to the regulations of a German teaching hospital, while also ensuring the viability of breaks. Currently, rosters are manually prepared by experienced physicians with basic spreadsheet knowledge, leading to significant costs and time consumption because of the complexity of the problem and the individual working conditions of the physicians. Unfortunately, manually generated rosters frequently prove to be non-compliant with labor regulations and ergonomic agreements, resulting in potential overtime hours and employee dissatisfaction. A particular concern is the inability of physicians to take mandatory breaks, which negatively affects both employee motivation and the hospital service level. To address these challenges, we propose a data-driven formulation of an operational physician scheduling problem, considering overstaffing and overtime hours as primary cost drivers and integrating shift preferences and break viability as ergonomic objectives. We develop and train a survival regression model to predict the viability of breaks, allowing practitioners to define break-time windows appropriately. Given the limitations of standard solvers in producing high-quality solutions within a reasonable timeframe, we adopt a Dantzig–Wolfe decomposition to reformulate the proposed model. Furthermore, we develop a branch-and-price algorithm to achieve optimal solutions and introduce a problem-specific variable selection strategy for efficient branching. To assess the algorithm's effectiveness and examine the impact of the new break assignment constraint, we conducted a comprehensive computational study using real-world data from a German training hospital. Using our approach, healthcare institutions can streamline the rostering process, minimize the costs associated with overstaffing and overtime hours, and improve employee satisfaction by ensuring that physicians can take their legally mandated breaks. Ultimately, this contributes to better employee motivation and improves the overall level of hospital service.

1. Motivation and introduction

In recent decades, hospitals have experienced a surge in patient care demands. This growth can be attributed to two main factors: a reduction in the number of hospitals [1] and the demographic shift, which is continuously leading to an increase in the number of patients due to the aging population [e.g., 2]. In Germany, the number of patients increased almost 25% within 25 years until 2015 [1] and the number of hospitals decreased from 2411 hospitals to 1951 hospitals during the same timescale [1,3]. To cope with this increasing workload, hospital management must find ways to efficiently manage medical personnel. Unfortunately, the significant number of vacant annual positions [4] prohibits the hiring of additional physicians, making it imperative for hospitals to optimize their existing resources and capacities.

Efficient personnel scheduling is vital to avoid additional overtime hours for physicians while ensuring the required quantity and desired quality of care. However, the current practice of creating schedules is manual, with experienced physicians using basic spreadsheet knowledge [5]. This process is time-consuming and costly due to the intricate nature of the problem and the individual working conditions of physicians. The workforce typically comprises a mix of residents and specialists, some of whom work part-time and others full-time. Additionally, the planner must deal with different input plans, further complicating practical implementation. For example, medical residents often have long-term schedules generated at least once a year, specifying the department they will work in Kraul [6]. On the other hand, the duty schedule is created monthly, assigning physicians to overnight duties [7]. Combining the information from these two schedules, the

☆ Area: Production Management, Scheduling and Logistics. This manuscript was processed by Associate Editor Kress.

* Corresponding author.
E-mail address: s.kraul@vu.de (S. Kraul).

https://doi.org/10.1016/j.omega.2024.103154
Received 18 September 2023; Accepted 12 July 2024
Available online 19 July 2024

responsible planner generates weekly rosters for each physician in the department between one and two months in advance. The rosters must be matched to a task schedule such as a surgery schedule, which is not sufficiently determined when planned [8]. Unfortunately, the resulting roster is often flawed and causes several problems, including overtime, understaffing, and insufficient breaks. These problems have direct implications on the department's service level and employee satisfaction, and overtime is a significant cost driver [9].

Consistent and anticipated breaks play an important role in improving the mental and physical health of physicians, which in turn impacts their effectiveness and the standard of care they provide. Physicians who are overburdened or take breaks inconsistently are more prone to increased stress and exhaustion. Consequently, this can result in reduced cognitive function, affecting the precision of diagnoses and the quality of patient care. Facilitating regular breaks for physicians also reflects ethical concerns about maintaining a healthy work-life balance. Physicians who can better anticipate their workload are likely to experience higher job satisfaction and reduced turnover rates, ultimately benefiting the healthcare system in general.

To address these challenges, the purpose of this paper is to determine an optimal work schedule for each physician covering a planning horizon of multiple weeks. We propose a mathematical programming approach that considers legal, ergonomic, and personnel-related aspects. This includes adhering to predefined start time windows for assigned shifts within a sequence of working days for each physician, minimizing overtime hours, and taking into account individual shift type preferences.

One of the main contributions of this research is the explicit planning and assignment of rest periods within physician shifts. We address the issue of current practices and literature neglecting to assign breaks to physicians, and we use a data-driven scenario-based approach to ensure the viability of breaks. This approach considers the fact that physicians may not always be able to take their breaks, despite legal requirements. We propose a survival regression model to define resilient break periods using surgery data from a full-care provider hospital with more than 1500 beds, allowing healthcare managers to determine break windows effectively. Additionally, the paper strives to accurately meet demand in each period of each planning day by minimizing the number of overstaffed periods within the planning horizon. We emphasize the practical applicability of our approach, making realistic assumptions to ensure its relevance in real-life settings. Furthermore, we highlight the potential for our approach to improve physicians' work-life balance by increasing planning certainty, allowing for solving a planning horizon consisting of multiple weeks. To address the formulated mixed-integer programming (MIP) model, we use a Dantzig–Wolfe decomposition and apply a branch-and-price (B&P) algorithm to generate individual schedules for each physician. We develop a problem-specific variable selection strategy for branching to enhance the efficiency of the algorithm.

The structure of the paper is organized as follows: The following section reviews the current literature on physician scheduling and related research aspects. Section 3 presents the problem and the formulated mathematical programming approach. Section 4 covers the Dantzig–Wolfe decomposition, the B&P approach, and the survival regression model. The experimental study is then presented in Section 5, evaluating the predictions of our survival regression model and the algorithm's performance, and analyzing the effect of considering the viability of breaks. Finally, the article concludes with a summary of the key insights and ideas for future research in Section 6.

2. Review of relevant literature

To pinpoint the research gap that we aim to address, it is necessary to examine various research streams. Initially, we will explore the most relevant literature on physician scheduling. Subsequently, we will evaluate the body of literature that considers breaks in staff scheduling problems. Lastly, we will examine the literature that delves into fairness and ergonomic aspects.

2.1. Physician scheduling

In recent decades, the problem of scheduling physicians has attracted increased attention. However, there is a notable disparity between the amount of research on this problem and the nurse scheduling problem. This section provides an overview of the relevant literature in current research. For a comprehensive review of the physician scheduling problem, we refer the interested readers to Erhard et al. [10].

In the current literature, physician scheduling problems typically involve work schedules that span predetermined planning horizons. However, the current literature has largely neglected the proper allocation of breaks for physicians. Baum et al. [11] scheduled physicians in a radiology division over a three-month planning horizon, focusing solely on predefined shifts. The incorporation of hospital-specific constraints resulted in an approximately 8% increase in revenue due to reduced paid working hours. Shamia et al. [12] also used predetermined shifts, employing a goal programming approach to assign physicians of varying experience and skills to shifts. Their objective was to maximize adherence to physician preferences, although optimizing resource utilization to improve service quality. The resulting mathematical model, solvable to optimality, delivered improved schedules in a short computation time. The flexibility in scheduling was increased by Bowers et al. [13], who proposed an MIP model that accounts for multiple shifts with varying start times, lengths, and desirabilities as defined by physicians. The MIP approach aimed to enhance the fairness of workload among physicians, achieving an average 7% fairness improvement, subsequently implemented in real-life scenarios.

Marchesi et al. [14] propose a two-stage stochastic model using a sample average approximation to assist in physician staffing and scheduling within an emergency department, taking into account uncertainties related to patient arrivals. They integrate various shifts and skill levels of physicians and aim to minimize the total expected number of patients waiting for treatment. Their findings indicate that their model can decrease the queue frequency by 22% based on a case study. The primary focus of Tohidi et al. [15] is on the uncertainty of demand with respect to treatment time. They employed a two-stage formulation, with the first stage characterized as an adjustable robust scheduling problem. Their objective is to minimize the costs associated with various resources, such as physicians and rooms, in addition to the costs of patient rejection. Through a computational analysis, they demonstrated that their modeling strategy leads to notably lower costs compared to its deterministic counterpart. Investigating physician rescheduling in conjunction with minimizing the risk tolerance level, Wang et al. [16] introduced a bi-objective formulation that balances operational costs and capacity shortage risk. They introduced an exact iterative algorithm capable of identifying all Pareto-optimal solutions.

All of these articles share the commonality of overlooking breaks during a work shift. Although the latest research acknowledges uncertainties regarding demand, the rosters they suggest are expected to underestimate the costs resulting from missed breaks. Furthermore, most studies focus on reducing operational costs, some incorporating a patient-centered approach. Although these points of view are crucial and deserve attention, a third aspect is absent, that of physicians. Given the significant staff shortage in the healthcare industry, hospital administrators must address this issue by improving the quality of work.

2.2. Planning of breaks

The current literature predominantly ignores break assignments in physician scheduling. Focusing on a physician staffing problem, Brunner et al. [17] introduced maximal flexibility in shift types, allowing different start periods and lengths. Break assignment and balanced shift starts were addressed within a mathematical programming framework solved using a decomposition heuristic. High-quality schedules were quickly produced by the heuristic approach, validated in real-world settings. To deal with larger instances, Brunner et al. [18] employed a B&P approach for optimality, improving performance in various test cases. Similarly to Brunner et al. [17,18], Erhard [19] introduced flexibility, integrating various sequences of working days, shift types, and break assignments, using a column generation (CG) heuristic due to the complexity of the problem. The results highlighted the substantial positive impact of flexibility on labor costs.

Stolletz and Brunner [20] analyze a physician scheduling problem that minimizes labor costs using a reduced set covering approach, demonstrating the superiority of explicit shift modeling over implicit assignment. They incorporate predefined break-time windows to ensure labor regulations and include fairness by minimizing both overtimes and undertimes of physicians regular working time.

In particular, breaks have received limited attention in all personnel scheduling literature [21]. Breaks are usually limited to predefined time windows [22], although Bechtold and Jacobs [23] extended this to the implicit formulation. Aykin [24] further enhanced the implicit formulation by accommodating three different breaks within one shift, reducing the variable count. Sungur et al. [25] adopted this approach to analyze optimal break periods and intervals, striving to minimize labor costs. Kiermaier et al. [26] introduced a two-stage decomposition to solve the assignment problem of breaks independently of the rostering problem. They proofed several break assignment problems to be NP-hard.

All of these papers assume that breaks are given as specific timeslots or time windows, without discussing and explaining how they are derived. In our paper, we close this gap by providing this sequential step in the modeling of breaks. We adopt the concept of a predefined break-time window using the earliest and latest break periods. Instead of a continuous time window, we further restrict the possibility of breaks. In addition, we propose a survival regression model to derive the break-time windows using real data to ensure more resilient scheduling.

2.3. Fairness and ergonomic aspects

In recent years, there has been a growing emphasis on integrating individual preferences into personnel scheduling problems [21]. Within any given organization, employees may find themselves assigned tasks or shifts that are not their top choices. Facilitating an even distribution of less preferred tasks or shifts is commonly viewed as a fair approach to personnel decisions [7,20,27–30]. An equal outcome is frequently achieved by incorporating soft constraints into multi-objective optimization processes [21,31].

Bard and Purnomo [32] responded to the shortage of nurses by creating a model that considers shift preferences, identifying different degrees of violation. They emphasize the importance of managing individual preferences and requests for days off in a manner that is considered fair. The model incorporates penalties for working overtime and for variations in shifts. The authors suggest a column generation heuristic to address their problem for a maximum of 100 nurses, incorporating both 8 and 12 h shifts. In pursuit of long-term fairness and adherence to individual preferences, Gross et al. [33] introduced an indicator that measures the fulfillment of individual preferences in independently generated one-month work schedules. The results of the formulated MIP model indicate a better performance of the implemented preference indicator in terms of an equal distribution of satisfaction between all physicians.

Adams et al. [34] aim to balance the workload among physicians by minimizing the largest difference in workload between several groups of physicians over the planning horizon. They generate cyclic rosters for general medical physicians to improve continuity of care.

The study conducted by Rea et al. [35] examines the balance between equality and equity in addressing fair personnel scheduling problems, using a case study on physician scheduling. Equity is defined by a variety of employee factors, such as seniority, leadership, productivity, and developmental efforts. A parameter is employed to allow management to assess schedules with varying emphases, thereby showcasing the Pareto frontier of the equity-equality trade-off. Through the integration of surveys, they demonstrated that employees' perceptions of fairness and satisfaction saw enhancements compared to the prior planning method.

The concept of fairness is a common topic in academic discussions. However, it is frequently highlighted that the perception of fairness may not align with the actual fairness in a given situation. Within a workday, fairness is determined by how shifts are allocated or the distribution of preferred tasks. Weekly or monthly, fairness is often described considering deviations from agreed-upon work hours, differences in shift patterns, and individual preferences for specific shifts. In this study, we will focus on factors such as shift preferences, deviations from contracted hours, and variations in shift schedules. Furthermore, we will introduce the viability of breaks as an additional criterion for fairness, a dimension that has not been explored in previous research.

To our knowledge, the existing literature lacks an approach that addresses the explicit placement of breaks at the operational level when creating work schedules for physicians with the uncertainty of taking a break. The paper closest to ours is from Stolletz and Brunner [20]. The main differences are that we include additional ergonomic measures, i.e., differences in start times of shifts within a week and viability of breaks. Moreover, we do not assume that breaks can be taken throughout a given time window. We describe and analyze a way to define appropriate break-time windows using survival regression models. Furthermore, this study aims to bridge research gaps by accommodating various groups of physicians who differ in labor contract, expertise, and shift preferences.

3. Problem statement and mathematical model

The problem under consideration aims to minimize overstaffing, overtime, and shift starting disparities, although concurrently maximizing physician preferences' adherence and break scheduling. To achieve this, we develop a distinct work schedule for each physician indexed by i, encompassing the entire planning horizon. The set of physicians, denoted as I, encompasses a range of employees distinguished by their contractual working regulations and qualification levels. Specifically, I^{spec}, a subset of I, pertains to physicians who have completed their residency programs. The planning period spans $|D|$ days, subdivided into $|P|$ periods of uniform duration, where each period p is defined as one hour. It is worth noting that alternative period lengths, such as 30-minute intervals, can be considered if required.

To establish individual rosters for each physician, a set of shifts denoted as S is available for assignment. The binary parameter A_{sp} defines the working periods within a shift, taking the value of 1 if the period p is part of a working period in shift s, and 0 otherwise. The binary decision variable x_{isd} is used to assign a specific shift s to a physician i on day d, with a value of 1 indicating assignment and 0 indicating no assignment.

It is important to consider physician availability, denoted by V_{id}, as not all physicians are available every day. Our focus aligns with rostering at an individual level, as presented by Erhard et al. (2017). This includes the inclusion of overnight duties, involving 24-hour shifts. In such cases, a physician, when assigned to duty on day d, first performs a 12 h shift, followed by 12 h on call during the night. Consequently, they are not assigned to any shift on the following day $d+1$. It is noteworthy that different duty structures are feasible, such as an eight-hour shift followed by 16 h on-call. Duty assignments are predetermined and serve as input, indicated by the binary parameter K_{id}, where 1 denotes the physician's duty on day d, and 0 indicates otherwise.

Table 1
Shift-specific break windows for one physician i and day d, 0 indicating the break.

$A_{sp} - b_{ispd}$	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	1	1	1	0	1	1	1	1				
2		1	1	1	1	0	1	1	1	1			
3					1	1	0	1	1	1	1	1	1
4	1	1	1	1	1	0	1	1	1	1	1		
5			1	1	1	1	0	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	0	1	1	1	1

To promote balanced shift assignments, we introduce the concept of time windows for shift starts. For a given physician i in week w, e_{iw} and l_{iw} represent the earliest and latest allowed shift start times, respectively. According to legal mandates, consecutive shift assignments are spaced by a minimum number of rest periods, denoted B^{rest}. Furthermore, we are bound by government and labor regulations, mandating compliance with a weekly working period limit ω_{iw} for each physician i, depending on their individual labor contracts.

To manage break assignments, a defined time window governs appropriate break periods. The parameter B_s^{pre} specifies the count of working periods within a shift preceding a break assignment, whereas B_s^{post} determines the minimum mandatory working hours after a break is assigned. Breaks are allocated using the binary decision variable b_{ispd}, which individually assigns breaks to physicians during specific shifts s, in periods p, and on days d.

For illustration, consider Table 1, which presents an example of break time windows for various types of shifts. This table includes six distinct shifts, each with varying start periods and durations. These shifts are available for personnel assignment across 13 planning periods within a single hour-long day.

The binary parameter A_{sp} delimits the working periods within a specific shift by taking the value of 1. For example, in the case of Shift 2, it starts in period 2 and ends in period 10. A sequence of 1s is interrupted by a 0, which signifies the allocation of a break during that shift, i. e., $b_{i,2,6,d} = 1$. Each shift has its own distinct break window, highlighted in gray. Examining shift 5, its break window spans from period 5 (determined by $B_5^{pre} = 2$, considering the shift's first working period in Period 3) to Period 10 ($B_5^{post} = 3$). The specific break for this shift is scheduled in period 7 as denoted by $b_{i,5,7,d} = 1$. It is important to acknowledge that these shift-specific break windows may be defined by the hospital administration, possibly based on factors like shift duration, with shorter shifts having shorter break windows and longer shifts having extended ones.

In practical terms, it is implausible for a physician to have an equal probability of taking a break in each slot of the break window. Factors such as the length of the surgery or the levels of demand can influence this viability. As a remedy, we introduce the concept of a break assignment viability o_{spd}^n, which is specific to the period, day, and shift, considering a predefined set of scenarios denoted as N. The set of indices considered may vary depending on the context. For instance, the viability could be unrelated to the shift if the responsibilities are shared among all staff members. On the other hand, the viability may be linked to the physician if the tasks that could result in a break are influenced by the physician. To improve resilience, a safety threshold α is defined to ensure a minimal level of break viability across all scenarios. This approach promotes a more reliable placement of breaks, increasing the likelihood that physicians can fulfill their mandatory breaks. The number of scenarios taken into account is an important factor and depends on the type of application. For example, variations in a surgical schedule are limited on a two-week horizon, so that the number of scenarios can be small. For applications where services are unlikely to change at all, scenarios might be omitted. A more detailed explanation will follow upon the introduction of the mathematical formulation. The resulting work schedules are structured on the basis of a diverse set of labor and governmental regulations. These work schedules are allocated in a way that guarantees coverage of both total demand R_{pd}^{total} and specialist demand R_{pd}^{spec} during each period p on all days d within the planning horizon.

In the following, we present our MIP model designed to generate personalized work schedules while also considering the viability of breaks. As our formulation addresses the optimization of physician rosters, the objective function is made up of two components. Cost-related measures accounting for an efficient matching of supply and demand and physician-related measures ensuring personnel satisfaction and job motivation. Therefore, the following notation is introduced:

Sets with indices

$i \in I$ set of physicians
$i \in I^{spec}$ set of specialists ($I^{spec} \subset I$)
$w \in W$ set of weeks
$d \in D$ set of days
$d \in D_w$ set of days within week w
$p \in P$ set of day-periods each 1 hour long
$s \in S$ set of shifts
$n \in N$ set of scenarios for the viability of a break

Parameters

R_{pd}^{total} total demand in period p on day d
R_{pd}^{spec} demand for specialists in period p on day d
A_{sp} 1 if shift s covers periods p, 0 otherwise
V_{id} 1 if physician i is available on day d, 0 otherwise
K_{id} 1 if physician i is on duty on day d, 0 otherwise
ω_{iw} number of working periods for physician i in week w
H_s length of shift s (working hours in periods)
F_s first working period of shift s
L_s last working period of shift s
B_s^{pre} minimum number of working periods before the break is allowed to start
B_s^{post} minimum number of working periods after the break ended
B^{rest} minimum number of rest periods between two consecutive shift assignments
c_{is}^{shift} preference score of physician i for shift s
c^{over} penalty term for overstaffing in a period
c^{time} penalty term for deviation in shift start time window
c^{hour} penalty term for overtime for a physician
o_{spd}^n viability of taking a break in shift s in period p on day d in scenario n
ρ_n probability of scenario n
α average day specific safety level for break viability

Decision variables

x_{isd} 1 if physician i is assigned to shift s on day d, 0 otherwise
b_{ispd} 1 if physician i working shift s on day d gets a break assigned in period p, 0 otherwise
e_{iw} earliest shift start of the physician i in week w
l_{iw} latest shift start of the physician i in week w
Δ_{iw}^{time} deviation from start time window constraints for the physician i in week w
Δ_{iw}^{hour} overtime of physician i in week w
Δ_{pd}^{over} overstaffing in period p on day d

$$\min \sum_{p\in P}\sum_{d\in D} c^{over}\Delta^{over}_{pd} + \sum_{i\in I}\sum_{w\in W} c^{hour}\Delta^{hour}_{iw}$$
$$+ \sum_{i\in I}\sum_{w\in W} c^{time}\Delta^{time}_{iw}$$
$$- \sum_{i\in i}\sum_{s\in S}\sum_{d\in D} c^{shift}_{is} x_{isd}$$
$$- \sum_{i\in I}\sum_{s\in S}\sum_{p\in P}\sum_{d\in D}\sum_{n\in N} \rho_n o^n_{spd} b_{ispd} \tag{1a}$$

s.t.

$$\sum_{i\in I}\sum_{s\in S}(A_{sp}x_{isd} - b_{ispd}) - \Delta^{over}_{pd} = R^{total}_{pd} \qquad \forall p\in P, d\in D \tag{1b}$$
$$\sum_{i\in I^{spec}}\sum_{s\in S}(A_{sp}x_{isd} - b_{ispd}) \geq R^{spec}_{pd} \qquad \forall p\in P, d\in D \tag{1c}$$
$$\sum_{s\in S} x_{isd} \leq V_{id} \qquad \forall i\in I, d\in D \tag{1d}$$
$$\sum_{p=F_s+B^{pre}_s}^{L_s-B^{post}_s} b_{ispd} = x_{isd} \qquad \forall i\in I, s\in S, d\in D \tag{1e}$$
$$\sum_{p=F_s+B^{pre}_s}^{L_s-B^{post}_s}\sum_{n\in N} \rho_n o^n_{spd} b_{ispd} \geq \alpha x_{isd} \qquad \forall i\in I, s\in S, d\in D \tag{1f}$$
$$\sum_{s\in S}\sum_{d\in D_w} H_s x_{isd} - \Delta^{hour}_{iw} = \omega_{iw} \qquad \forall i\in I, w\in W \tag{1g}$$
$$|P| - \sum_{s\in S} L_s x_{isd} + \sum_{s\in S}(F_s - 1)x_{isd+1}$$
$$\geq B^{rest}\sum_{s\in S} x_{isd+1} \qquad \forall i\in I, d\in D\setminus\{|D|\} \tag{1h}$$
$$l_{iw} \geq \sum_{s\in S} F_s x_{isd} \qquad \forall i\in I, d\in D_w, w\in W \tag{1i}$$
$$e_{iw} \leq |P| - \sum_{s\in S}(|P| - F_s)x_{isd} \qquad \forall i\in I, d\in D_w, w\in W \tag{1j}$$
$$m_{iw} - e_{iw} \leq \Delta^{time}_{iw} \qquad \forall i\in I, w\in W \tag{1k}$$
$$x_{i|S|d} \geq K_{id} \qquad \forall i\in I, d\in D \tag{1l}$$
$$x_{isd}, b_{ispd} \in \{0,1\} \qquad \forall i\in I, s\in S, p\in P, d\in D \tag{1m}$$
$$\Delta^{over}_{pd}, \Delta^{hour}_{iw}, \Delta^{time}_{iw}, e_{iw}, l_{iw} \geq 0 \qquad \forall i\in I, p\in P, d\in D, w\in W \tag{1n}$$

The objective function (1a) comprises five different parts: The first term seeks to minimize the total cost associated with overstaffed periods throughout the planning horizon. Parts two and three collectively minimize instances of overtime hours for physicians, although ensuring compliance with predefined time window constraints for each physician within each sequence of working days. The fourth part of the objective function is dedicated to maximizing the overall shift preference score. The final term is geared towards ensuring the highest attainable level of break assignment viability.

It is worth emphasizing that this paper's primary focus lies in proposing novel data-driven break window designs, rather than extensively delving into multiobjective optimization. Therefore, the balance and weighting of these objectives must be thoughtfully tailored to practical considerations. The range of all our objectives is as follows: Overstaffing (Δ^{over}_{pd}) varies from 0 to $|I| - R^{total}_{pd}$ for each period p and day d. Overtime hours (Δ^{hour}_{iw}) are within the range of $[0, |D_w| \max_{s\in S}\{H_s\} - \omega_{iw}]$ for each physician i and week w. Starting times (Δ^{time}_{iw}) can fluctuate between 0 and $\max_{s\in S}\{F_s\} - \min_{s\in S}\{F_s\}$ for each physician i and week w. For the shift preferences, at most one shift can be selected per day. The last term encompassing the viability of break assignments spans from $[\alpha|N|, |N|]$.

The initial set of Constraints (1b) and (1c) addresses the imperative of fulfilling demand requirements for each period within the planning horizon. Tailored to the level of expertise and the ensuing task capabilities, a distinction is made between "regular" physicians/residents and "specialists", i. e., the collective workforce and physicians who have successfully completed their residency program. Consequently, two distinct categories of demand constraints are required. The formulation can be easily extended to accommodate additional levels of expertise.

Constraints (1b) ensure that regular demand is met in every period of each day, excluding any instances of understaffing. The left-hand side of these constraints aggregates all physicians assigned to shifts during each period. As breaks are also assigned to each physician, the deduction of physicians taking breaks within a specific period is incorporated, represented as $\sum_{i\in I}\sum_{s\in S}(A_{sp}x_{isd} - b_{ispd}), \forall d\in D, p\in P$. Furthermore, an allowance for overstaffing, denoted as Δ^{over}_{pd}, is introduced in each period to account for situations where precise demand fulfillment is infeasible.

Constraints (1c), similar in structure to Constraints (1b), refer to the demand of specialists. In this case, only a lower bound is stipulated, allowing for surpassing it without incurring penalties in the objective function. This minimum number of specialists is vital because of their roles in supervision and handling complex tasks.

Constraints (1d) secure the assignment of shifts exclusively to available staff. In other words, if a physician i is available on day d, indicated by the parameter $V_{id} = 1$, the shifts can only be assigned to physician i for day d.

The subsequent set of Constraints (1e) and (1f) is dedicated to the allocation of breaks for physicians working on a given shift s on day d. Constraints (1e) ensure that a break is assigned for every physician working on a shift, meaning that when $x_{isd} = 1$, then $\sum_{p=F_s+B^{pre}_s}^{L_s-B^{post}_s} b_{ispd} = 1$ must hold. Here, the placement of the break is endowed with flexibility within a predefined time window specified by parameters B^{pre}_s and B^{post}_s. The parameter B^{pre}_s dictates the minimum number of working periods before the break assignment for a particular shift s, whereas B^{post}_s guarantees the minimum number of working periods after the break assignment. As the rigid block structure for breaks, as depicted in Table 1, does not necessarily mirror practicality (i.e., it might be plausible to take a break in period 6 for shift 5 but not in period 7), Constraints (1f) are introduced to provide a more realistic depiction of feasible break periods.

Table 2
Example of breaks adherence for shift type 1 using viability scores.

n\p	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0.701	0.677	0.75	0.608	0.663	0	0	0	0	0	0
2	0	0	0.737	0.936	0.644	0.918	0.899	0	0	0	0	0	0
3	0	0	0.897	0.633	0.832	0.608	0.854	0	0	0	0	0	0
Avg.	0	0	0.779	0.749	0.742	0.711	0.805	0	0	0	0	0	0

These constraints incorporate a safety level α to ensure the practicability of break assignments that function as a lower bound. This type of constraint is largely driven by data. The left-hand sides of these constraints aggregate the viability scores of adhering to a break assignment in all scenarios in N for each period within shifts. This sum is mandated to be at least as high as the designated α value. In this way, we can ensure that every physician has the specified minimum viability score of taking a break.

Consider the following example to illustrate this concept. Assume $|N| = 3$, and the viability scores for taking a break for a shift beginning in period 1 and ending in period 9 (as detailed in Table 1) are presented in Table 2. For simplicity, we assume that the probabilities of each scenario (ρ_n) are the same. Applying a safety level of $\alpha = 0.75$ would result in feasible breaks occurring in periods 3 or 7. The viability scores for the assignment of breaks can be derived using a survival regression model, as described in Section 4.2, to integrate real-world data in the calculation of viability scores. Note that this step of evaluating the feasible break periods can be performed in a preprocessing.

Constraints (1g) and (1h) are instrumental in the management of governmental and labor-related working regulations. Constraints (1g) ensure that maximum weekly working periods for personnel are not exceeded. This consideration is crucial as we tackle a rostering problem, accounting for individual working hour regulations ω_{iw}, applicable for all physicians i within a set W. It is noteworthy that working hours are contingent on the set W as it needs to accommodate vacation and absences in ω_{iw}. In pursuit of efficient alignment between demand and supply, we exclusively allow weekly overtime periods Δ^{hour}_{iw} for each physician. These overtime periods are subjected to penalties within the objective function. Constraints (1h) require a minimum number of rest periods B^{rest} between two consecutively assigned shifts for each physician. As this minimum rest interval is a legal requirement, the left-hand side of these constraints enumerates the number of off periods between the last working period L_s of one shift on day d and the initial working period F_s of a shift on the consecutive day $d+1$. This count of off periods must be equal to or exceed the value specified on the right-hand side, i.e., B^{rest}. In particular, B^{rest} requires customization based on the set and dimension of P, for example, accounting for one-hour periods versus 15-minute intervals.

Moving to Constraints (1i) through (1k), this set defines the scope for starting periods of consecutive shift assignments within a week, encompassing a sequence of working days. Constraints (1i) and (1j) determine the latest and earliest allowable shift start times for one week. Subsequently, Constraints (1k) establish the range within which the start time window of the shift must be confined, as represented on the left-hand side of the constraints. Deviations from this range are prohibited.

Constraints (1l) are formulated to handle overnight duty assignments. These duties encompass a specific regular shift followed by an on-call duty. If a physician is assigned an overnight duty on day d, these constraints ensure appropriate shift assignment, specifically the last shift in the set S. It is important to note that for physician i, $V_{id+1} = 0$ if $K_{id} = 1$, indicating unavailability on the subsequent day.

Ultimately, Constraints (1m) and (1n) define the domains of the variables. It is pertinent to mention that the continuous variables will be integer in a feasible solution.

4. Branch-and-price algorithm and survival regression models as a solution approach

4.1. Branch-and-price algorithm

Model (1), which presents the compact formulation of the rostering problem, encounters difficulties in producing feasible meaningful solutions within a realistic timeframe using standard solvers. Consequently, we have devised an alternative strategy to address this challenge in this section.

Taking into account the inherent block structure of the problem, a promising approach emerges: the adoption of a Dantzig–Wolfe reformulation, as proposed in prior research [36–38]. Leveraging this concept, we embark on the decomposition of the compact rostering problem (1) through a Dantzig–Wolfe reformulation. Subsequently, we proceed to construct a B&P algorithm, designed to achieve optimality by solving this reformulated problem.

We decompose our model by physicians, acknowledging the inherent heterogeneity among them in terms of working hours, availability, and overnight duties. This precludes us from aggregation. The compact formulation is decomposed into a master problem (MP) and $|I|$ subproblems (SP(i) $\forall i \in I$). Given that not all columns of the MP are typically known in advance, a restricted MP is formulated, which includes a subset of columns. In the following, the restricted MP will be labeled as MP. In an iterative manner, the SPs are utilized to generate new columns, with the dual solution of the MP steering this generation process.

Decomposition of the rostering problem. For the formulation of the MP, a new set of columns $K_i = \{1, \dots, |K_i|\}$ must be introduced (representing extreme points). Every $k \in K_i$ specifies a feasible roster for a physician i reserving individual working restrictions and overnight duties. Note that the set K_i is increasing during the solution process as new rosters are generated in the SPs. Additionally, our reformulation does not need extreme rays because SP(i) is bounded [36]. The resulting MP (extensive formulation) of our rostering problem can be formulated as follows:

Additional sets with indices

$k \in K_i$	set of rosters for physician i

Additional parameters

X^k_{pd}	1 if roster k covers period p on day d, 0 otherwise
$c^{pattern}_k$	total cost of the roster k (comprising costs derived from overtime periods, time window and shift preference violations, and break viability)

Additional decision variables

λ_{ik}	1 if physician i is assigned to the roster k, 0 otherwise

$$\min \sum_{p\in P}\sum_{d\in D} c^{over}\Delta^{over}_{pd} + \sum_{i\in I}\sum_{k\in K_i} c^{pattern}_k \lambda_{ik} \tag{2a}$$

s.t.

$$\sum_{i\in I}\sum_{k\in K_i} X^k_{pd}\lambda_{ik} - \Delta^{over}_{pd} = R^{total}_{pd} \qquad \forall p\in P, d\in D \qquad (\Pi^{total}_{pd}) \tag{2b}$$

$$\sum_{i\in I^{spec}}\sum_{k\in K_i} X^k_{pd}\lambda_{ik} \geq R^{spec}_{pd} \qquad \forall p\in P, d\in D \qquad (\Pi^{spec}_{pd}) \tag{2c}$$

$$\sum_{k\in K_i} \lambda_{ik} = 1 \qquad \forall i\in I \qquad (\Pi^{physician}_{i}) \tag{2d}$$

$$\lambda_{ik} \in \{0,1\} \qquad \forall i\in I, k\in K_i \tag{2e}$$

$$\Delta^{over}_{pd} \geq 0 \qquad \forall p\in P, d\in D \tag{2f}$$

The objective function of the MP (2a) minimizes the total costs of overstaffed periods within the planning horizon in addition to the total costs of the assigned working patterns k for the entire workforce, i.e., $c^{pattern}_k$ contains overtime, time window, shift preference, and break viability costs. Constraints (2b) ensure demand to be covered in every period p of every day d within the planning horizon, whereas Constraints (2c) account for a minimum number of specialists in each period p of every day d. The parameter X^k_{pd} is analogous to the compact formulation of $\sum_{s\in S} A_{sp}x_{isd} - b_{ispd}$ representing whether a physician is working in period p on day d or not. The values of parameter X are determined in the specific SP of physician i, i.e., SP(i), as detailed in the following paragraph. The convexity constraints for every physician i are given in Constraints (2d) whereas variable domain definition is defined in Constraints (2e) and (2f).

As the dual formulation of the MP does not provide relevant insights, only the dual variables Π^{total}_{pd}, Π^{spec}_{pd}, $\Pi^{physician}_{i}$ of Constraints (2b) to (2d) are detailed here. These will be used to determine the reduced cost of a generic MP column and are considered in the objective function of the SPs. Note that Π^{total}_{pd}, $\Pi^{physician}_{i} \in \mathbb{R}$, whereas $\Pi^{spec}_{pd} \in \mathbb{R}^+$. The generic reduced cost of a column (roster k of physician i) can be written as in (3).

$$c^{pattern}_k - \sum_{p\in P}\sum_{d\in D} X^k_{pd}(\Pi^{total}_{pd} + \Pi^{spec}_{pd}) - \Pi^{physician}_{i} \tag{3}$$

Note that Π^{spec}_{pd} is only part of the objective of SP(i) if physician i is a specialist, i.e., $i \in I^{spec}$. The solution of SP(i) can be interpreted as a feasible roster for the complete time horizon. The parameters and variables in the SP will be the same as in the compact formulation (1) of Section 3, except the dropping index i of the decision variables. With this in common, the SP(i) can be formulated as:

$$\begin{aligned}
\min \sum_{w\in W} c^{hour}\Delta^{hour}_{w} + \sum_{w\in W} c^{time}\Delta^{time}_{w} - \sum_{s\in S}\sum_{d\in D} c^{shift}_{is} x_{sd} \\
- \sum_{s\in S}\sum_{p\in P}\sum_{d\in D}\sum_{n\in N} \rho_n o^{n}_{spd} b_{spd} \\
- \sum_{s\in S}\sum_{p\in P}\sum_{d\in D} (A_{sp}x_{sd} - b_{spd})(\Pi^{total}_{pd} + \Pi^{spec}_{pd}) - \Pi^{physician}_{i} \qquad (4a)
\end{aligned}$$

s.t.

$$\sum_{s\in S} x_{sd} \le V_{id} \qquad \forall d \in D \quad (4b)$$

$$\sum_{p=F_s+B^{pre}_s}^{L_s-B^{post}_s} b_{spd} = x_{sd} \qquad \forall s \in S, d \in D \quad (4c)$$

$$\sum_{p=F_s+B^{pre}_s}^{L_s-B^{post}_s} \sum_{n\in N} \rho_n o^{n}_{spd} b_{spd} \ge \alpha x_{sd} \qquad \forall s \in S, d \in D \quad (4d)$$

$$\sum_{s\in S}\sum_{d\in D_w} H_s x_{sd} - \Delta^{hour}_{w} = \omega_{iw} \qquad \forall w \in W \quad (4e)$$

$$\begin{aligned}
|P| - \sum_{s\in S} L_s x_{sd} + \sum_{s\in S}(F_s - 1)x_{sd+1} \\
\ge B^{rest} \sum_{s\in S} x_{isd+1} \qquad \forall d \in D \setminus \{|D|\} \quad (4f)
\end{aligned}$$

$$l_w \ge \sum_{s\in S} F_s x_{sd} \qquad \forall d \in D_w, w \in W \quad (4g)$$

$$e_w \le |P| - \sum_{s\in S}(|P| - F_s)x_{sd} \qquad \forall d \in D_w, w \in W \quad (4h)$$

$$m_w - e_w \le \Delta^{time}_{w} \qquad \forall w \in W \quad (4i)$$

$$x_{|S|d} \ge K_{id} \qquad \forall d \in D \quad (4j)$$

$$x_{sd}, b_{spd} \in \{0,1\} \qquad \forall s \in S, p \in P, d \in D \quad (4k)$$

$$\Delta^{hour}_{w}, \Delta^{time}_{w}, e_w, l_w \ge 0 \qquad \forall w \in W \quad (4l)$$

As SP(i) searches for a new promising column in each iteration, the objective function (4a) minimizes the reduced cost of the corresponding column. Therefore, the dual values Π^{total}_{pd}, Π^{spec}_{pd}, and $\Pi^{physician}_{i}$ derived from Constraints (2b) to (2d) are in use. Thus, SP(i) generates individual work schedules that specify a sequence of shift and break assignments covering the entire planning horizon. The Constraints (4b) to (4l) of the SP correspond to the Constraints (1d) to (1n) of our compact formulation in Section 3. As these constraints are already discussed in detail in a previous section, we do not explain them any further.

In the sense of column generation, any new feasible roster can be integrated into the MP if and only if the objective value is smaller than 0, i.e., a negative reduced cost column can be generated. In this case, the values of $X^k_{pd} := \sum_{s\in S} A_{sp}x_{sd} - b_{spd}$ for all $d \in D, p \in P$ define a new column for a specific physician i that is added to the set K_i in MP. If the reduced costs are 0 for all SP(i), then no promising column exists and the column generation procedure terminates with the optimal linear programming solution for MP.

Finding integer solutions. To attain an optimal solution to the original problem, the column generation procedure must be embedded in a B&P framework, particularly when the λ variables lack integer values. The use of an appropriate strategy for branching is a crucial element. However, this aspect has been extensively explored in the existing literature, as discussed by Barnhart et al. [36], Vanderbeck [39], and Desrosiers and Lübbecke [40] among others.

In our approach, we have designed and tested a branching strategy rooted in the work of Ryan and Foster [41], which was subsequently extended by Vanderbeck and Wolsey [42]. In this strategy, we consistently choose a single row associated with Constraints (2b) (for

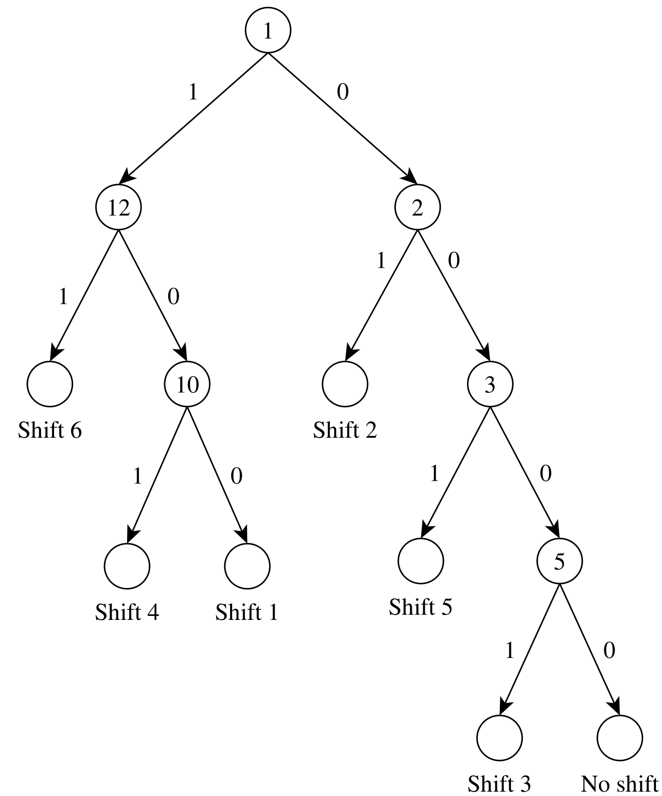

Fig. 1. Branching scheme based on the shifts of Table 1.

instance, row r) and another row linked with Constraints (2d) (for instance, row s). As expounded by Barnhart et al. [36], this selection yields a specialized branching scheme that naturally corresponds to the original formulation. In essence, the decision is made as to whether physician i should work in period p on day d or not. The resulting pair of branching constraints is given in (5).

$$\sum_{k\in K_i | X^k_{pd}=1} \lambda_{ik} = 1 \quad \text{and} \quad \sum_{k\in K_i | X^k_{pd}=1} \lambda_{ik} = 0 \qquad (5)$$

Note that the row r corresponds to a tuple of period p and day d and the row s to a specific physician i. Additionally, Constraints (2c) do not need to be considered in the branching scheme, because the values become integer as soon as the values in Constraints (2b) are integer.

To implement the strategy, the corresponding bounds on the branching variables need to be imposed at every node of the branching tree. Therefore, we first delete all patterns that do not satisfy the branching condition. Second, we force the corresponding SP to generate only patterns considering the branching decision. Therefore, we do not have to extend our MP with additional constraints. Moreover, the structure of the SPs is not changed because we only force a specific shift and break to be chosen, i.e., invalid shifts are deleted from the set S for a specific day d. Note that only for the left branch, i.e., $\sum_{k\in K_s | X^k_r=1} \lambda_{sk} = 1$, a physician is forced to work on a specific day.

In order to accelerate the branching scheme, we prioritize the examination of periods that identify distinct shifts. Subsequently, we address the intermediate periods that denote individual breaks within each shift. As we have more day periods than start periods of shifts, i.e., $|P| \gg |S|$, we identified a special selection to build a more balanced search tree for each physician with respect to the shifts. Continuing with the six shifts in Table 1, we evaluate for every day d and physician i whether $\sum_{k\in K} X^k_{pd}\lambda_{ik}$ is fractional or not in the order given in Fig. 1 – the number in the vertices corresponds to the period p. If the corresponding sum is fractional in one stage, we perform the

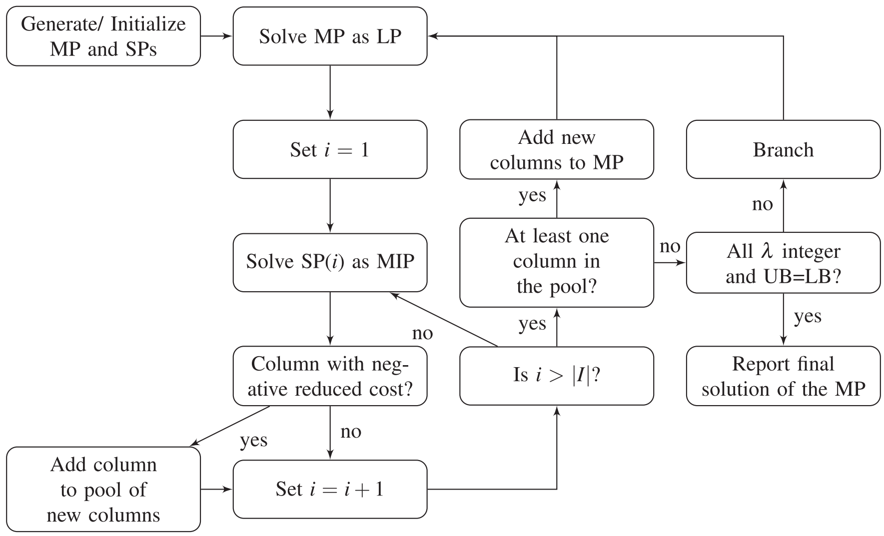

Fig. 2. Flow chart of the B&P framework.

branching in this period; we continue the tree otherwise. If there is no fractional value, then we start branching on the break periods of the remaining shift. Note that at most one shift is left when the branching scheme is used. For example, let $\sum_{k\in K_i} X^k_{pd}\lambda_{ik}$ equal one for $p = 1$ and equal zero in $p = 12$. In the next stage of the branching scheme we check the same sum for period $p = 10$. Let us assume this sum is fractional. By performing the left branch ($X_{10,d} = 1$) we force the algorithm to use shift 4. In the other branch ($X_{10,d} = 0$) only shift 1 can be used. Note that in each stage, we bisect the number of feasible shifts.

In addition to the branching scheme, node selection is another critical issue in developing a B&P algorithm. We implemented a depth-first strategy to identify an upper bound early. In this approach, each iteration involves the exploration of one of the successor nodes. In cases where multiple nodes are viable options, we resolve ties by opting for the node with the highest index. As a basic primal heuristic, we solve the MP with all columns found so far as an MIP, whenever the algorithm finds an integer solution. A flow chart of all the steps of the B&P framework can be seen in Fig. 2.

4.2. Survival regression model

Survival regression models play a fundamental role in statistical analysis when the focus is on the time until a specific event takes place [43]. This is especially common in fields like medical research, where the event could be mortality, recurrence, or, in our context, the completion of a surgical procedure or the presence of an anesthesiologist. The rationale for utilizing survival regression models is their ability to give the probability that an event occurred and to consider various variables that could affect the duration of survival. These models not only predict the survival function but also enable the integration of covariates, offering insights into how different factors influence the likelihood of survival as time progresses.

In a survival regression model, there are two primary components. First, the survival function $S(t)$ offers the probability of survival in a specific time period t. The second element is the hazard function denoted as $h(t)$, which indicates the rate at which the event occurs. Essentially, it represents the speed at which an object, which is free of the event at a particular time, will undergo the event. These two functions are interlinked, and the hazard function reflects the rate at which the survival function decreases over time. Applied to our case, we would like to measure the probability that the obligatory presence of an anesthesiologist will be expected at a certain point in time.

Accelerated failure time model. Depending on the assumptions made regarding the distribution of survival time or the hazard function, survival regression models can be categorized into non-parametric, semi-parametric, and parametric models, each with its own advantages and practical applications [44]. In our setting, parametric models are preferred because the duration of surgery can often be accurately estimated by fitting established distributions such as the normal or lognormal distribution [45,46]. Another area of research focuses on utilizing regression models for predicting surgery duration [47]. The accelerated failure time (AFT) model is a specific form of parametric survival regression model that closely resembles traditional linear regression models. The AFT model posits a direct link between predictors (input) z_j and survival time.

The AFT model can generally be represented as shown in Eq. (6), where the natural logarithm of the failure time t is a linear combination of the input variables z_j with coefficients β_j for all the inputs $j \in J$. Note that the failure time t can be projected in the periods $p \in P$ of the mathematical model (1). The error term ϵ is assumed to adhere to a particular parametric distribution. Note that the two key distinctions from conventional regression models, apart from censored observations, are that the input influences the event time multiplicatively and that the error terms are not normally distributed [43].

$$\ln(t) = \beta_1 z_1 + \cdots + \beta_{|J|} z_{|J|} + \ln(\epsilon) \tag{6}$$

Defining appropriate input variables. The input variables z_j play a critical role in predicting the duration of surgery. Since rosters are usually generated between one and two months in advance, the available input data are limited and subject to change, as planned (elective) patients may be rescheduled. Furthermore, the patient-anesthetist assignment is not (always) known at this planning stage. Therefore, we will utilize only the information present at the time of roster preparation.

A commonly used method in hospitals to estimate the duration of surgery is the Last-5-case approach, which indicates the average duration of the last five surgeries [43]. We will adopt this methodology and assess the effectiveness of various rolling horizons as input variables in

Table 3
Input data for the computational study.

Specialists/ residents (I)	Working time (ω)	Shifts (S)	Periods (P)	Scenarios (N)	Days (D)
10/10	$40h/24h$	6	13	15	7/**14**/21/28

our experimental study. We define the Last-γ-case as

$$\bar{M}_{q+\gamma+1} = \frac{\sum_{r=1}^{\gamma} M_{q+r}}{\gamma}, \tag{7}$$

where M_q is the duration the anesthetist attended the q's surgery and γ the number of previous surgeries that are taken into account. To enable the model to capture trends, we will incorporate not only the current average duration but also the average duration from the preceding surgery as input, i.e., $\bar{M}_{q+\gamma+1}$ and $\bar{M}_{q+\gamma}$.

In addition to details specific to the surgical procedure, we will provide information related to temporal and patient-specific information. Temporal details include the day of the week for surgery and the scheduled start time for the presence of the anesthesiologist. Patient-specific details cover age, sex, comorbidities, and the need for admission to the intensive care unit.

To address the various scales of the variables and enhance the model's convergence and interpretability, we normalize all input (and output) variables to a range of 0 to 1. Categorical data are processed using one-hot encoding. This resulted in a total of 12 inputs that can be combined to $|J| = 4095$ input variables.

5. Experimental study

In this section, we apply our model in a real-world case study concerning rostering of anesthesiologists for the operating theater of a German training hospital with more than 1500 beds. An overview of the input data for the computational study is given in Table 3. The study considers 20 physicians, i.e., $|I| = 20$. The workforce consists of 10 specialists and 10 residents. Additionally, 5 residents are part-time, i.e., with a weekly working hour of $\omega_{iw} = 24$. The remaining 15 physicians have a standard contract of 40 hours per week.

We have in total six different shifts as already presented in Table 1 of Section 3. To have a high flexibility, the number of working periods before and after a break is 2, i.e., $B_s^{pre} = B_s^{post} = 2$ for all shifts. Additionally, 15 scenarios are considered for breaks representing the probability that a physician can take a break during a specific period of a shift, i.e., $|N| = 15$. The scenarios are generated based on our AFT model fitted by surgery data from more than 100,000 surgeries and will be described in Section 5.1. A basic construction algorithm for generating scenarios artificially is stated in Appendix A. The weights for overstaffing (c^{over}), overtime (c^{time}), and variation in starting times (c^{start}) are set to 1 unless otherwise specified. This allows us to focus on the ergonomic aspects shift preferences and viability of breaks, without ignoring overstaffing.

All calculations are performed on a 2.7 GHz (Intel® Core™ i7-3740QM CPU) with 16 GB RAM running on the Windows 8.1 Enterprise operating system. The algorithm is coded in Python 3.5 and is linked to Gurobi 9.0 to solve the LP-relaxation of the MP and the MIP SPs. The default settings of Gurobi are used when solving MP and SP. SPs are always solved to optimality with a gap of $\epsilon = 0.001$, and only the optimal solution is transformed into a new column of the MP if the reduced costs are negative. The algorithm terminates when the optimal solution is found. In Section 5.2 we analyze our B&P algorithm for different problem sizes in terms of the lengths of the considered planning horizon. The real-world case is evaluated and discussed in Section 5.3.

5.1. Evaluation of the accelerated failure time model

The AFT model is designed to give us the probability that the presence of an anesthetist will no longer be required for a particular surgery. To achieve this, we conduct training on our AFT model using a data set that encompasses six years of surgical procedures at a hospital in Germany, involving a total of 3723 distinct types of surgery identifiable by their ICD codes, such as hip replacement surgery. The typical duration an anesthetist dedicates to a single surgery is 135 min on average, although this varies significantly depending on the type of surgery (see Fig. 3). The left part highlights the presence (in minutes) of anesthetists for all surgeries, and the right part highlights the presence of anesthetists for the most common type of surgery. Due to these variations, we developed a unique AFT model for each surgical type. The data set for each surgical type was divided into 20% for testing and 80% for training purposes.

Table 4
AFT model evaluation for the most common surgery.

Distribution	AIC	BIC	Concordance index
Weibull	−3157.22	−3190.97	0.69
Log-normal	−3081.10	−3114.85	0.69
Log–logistic	−3782.22	−3835.97	0.78

Alongside the error term, the input variables and their interactions significantly impact the efficacy of the AFT model. In a preliminary test, we could identify a rolling horizon of $\gamma = 3$ to be the most accurate for our approach. Throughout the process of refining our models, we were able to pinpoint the significant input factors for our models. Although we initially assumed that the day of the week would influence performance, our model evaluations revealed that the most accurate predictions are achieved with the inclusion of the following inputs: scheduled start time for the presence of the anesthesiologist (Hour), average duration of the Last-3-case (Rolling_avg) as well as the average duration of the Last-3-case for the previous surgery (Rolling_avg_prev), patient age (Age), number of comorbidities (Secondary), and planned admission to the intensive care unit (ICU). This resulted in a total number of input variables $|J| = 32$. A comprehensive description of our AFT model for the most common surgery can be found in Appendix B.

In Section 4.2, we described that the survival time in AFT models follows a particular parametric distribution. Our investigation involved testing three distinct distributions for our model: Weibull, Log–logistic, and Log-normal. We compare the performance of the three distributions using the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). The lower the value, the better the performance. Additionally, we used the concordance index to measure the model's ability to correctly rank pairs of surgeries based on their predicted survival times. The index can be between 0 and 1. A value of 0.5 is the expected result from random predictions and a value of 1 represents perfect concordance. Table 4 presents the evaluation of the three distributions for the most common surgery.

All models demonstrated good performance in ranking various surgeries, achieving concordance indices of 0.69 and 0.78. Upon comparing the models using their AIC and BIC scores, it becomes evident that the optimal selection is the log–logistic model. This holds true across various surgery types as well. The concordance index for the log–logistic AFT model varies for different surgeries, ranging from 0.74 to 0.81.

Trained AFT models can offer the planner the probability that an anesthetist is no longer required for a particular surgery. An illustration of the survival function based on our model is presented in Fig. 4. Combining this with the upcoming surgery schedule, we can determine

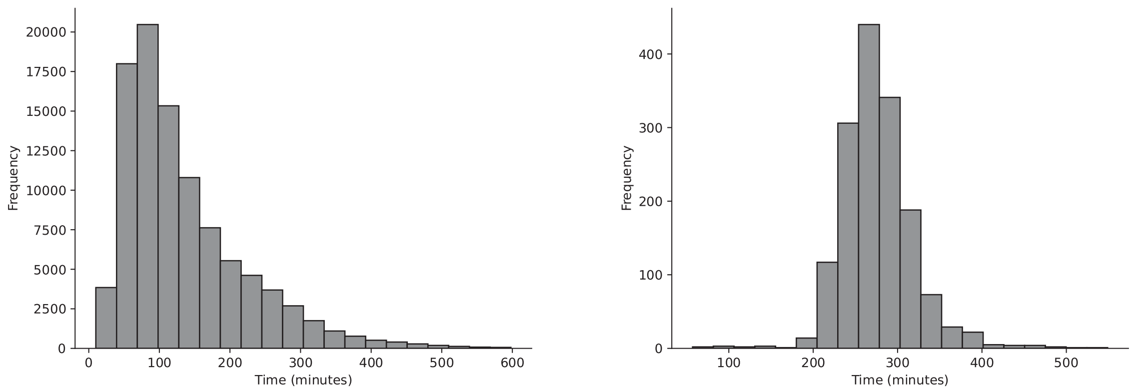

Fig. 3. Histograms of the presence of anesthetists. left: for all surgeries (N=114,164); right: for the most common type of surgery (N=1,558).

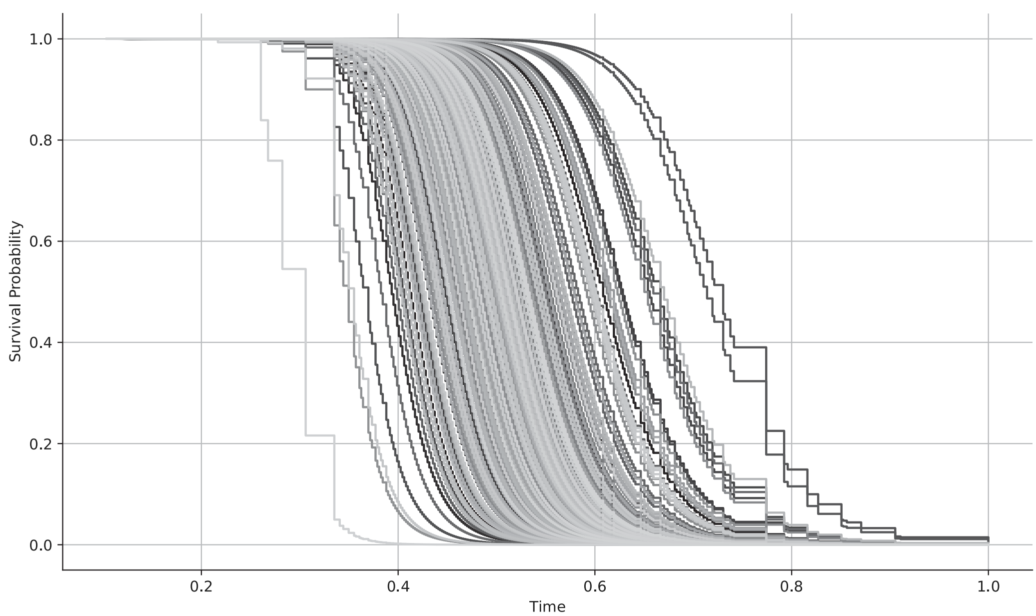

Fig. 4. Survival functions of the test set for the most common type of surgery (N=312).

the viability o^n_{spd} by adding the output of the AFT models for all surgeries that are active during the specific period p of a day d and dividing it by the number of active surgeries.

5.2. Evaluation of the branch-and-price algorithm

In order to evaluate the performance of the B&P algorithm, we test our model for different problem sizes. As the workforce is already provided on an operational level, we manipulate the size of the model by varying the time horizon of our roster, i.e., the number of days $|D|$. Note that a detailed overview of the input parameters is given in Appendix C. Although the standard time horizon of a roster is two weeks, we will evaluate the performance for 7, 14, 21, and 28 days. Although a four-week time interval seems attractive for physicians, schedules are often generated one week in advance due to high uncertainty [48]. For all time horizons, we perform ten runs using different shift preferences c_{is} for each physician i. Preferences are generated randomly for each instance using a Bernoulli distribution for every physician i and shift s [33]. In order to have an effect on the instances, the weight c_{is} of a preferred shift is set to 8 and 0 otherwise. The safety level α is set to 0.75. The aggregated results are given in Table 5. Note that we first list the average value over all ten instances followed by the maximum value.

The first column gives information on the time horizon, whereas the second column lists the GAP in percent. All 40 runs can be solved to optimality. The solution time of the MP in seconds is given in Column 3 whereas the average time of solving an MP once is given in Column 4 in milliseconds. Column 5 and 6 give the same information for the SPs. Comparing the solution time of the MP and SP we can see that the MP time is much more increasing than the SP time. For a time horizon of 7 days, only $0.4s$ are needed for the MP, whereas $3.4s$ are needed for the SPs. However, in a time horizon of 28 days, the time spent in the MP is $272.5s$, while only $150.0s$ is spent in the SPs. One

Table 5
Impact of different time horizons on the algorithm's performance ($N = 10$).

No. of days	GAP (%)	MP Time (s)	Avg. MP Time (ms)	SP Time (s)	Avg. SP Time (ms)	No. columns gen.	No. nodes expl.
7	0.0	0.4 | 0.5	11.5 | 14.4	3.4 | 3.6	5.8 | 6.1	585.5 | 588	12.7 | 13
14	0.0	9.0 | 13.7	71.2 | 84.8	18.8 | 26.5	10.1 | 12.2	1,812.6 | 2,215	33.9 | 57
21	0.0	112.4 | 142.3	292.9 | 316.3	81.0 | 93.8	15.5 | 16.4	5,232.0 | 5,995	94.6 | 112
28	0.0	272.5 | 347.7	512.4 | 557.9	150.0 | 208.6	22.4 | 26.4	6,680.1 | 7,889	123.8 | 148

Table 6
Impact of different weight factors on the algorithm's performance ($N = 10$).

Weights	GAP (%)	MP Time (s)	Avg. MP Time (ms)	SP Time (s)	Avg. SP Time (ms)	No. columns gen.	No. nodes expl.
1-1-1-1	0.0	9.0 | 13.7	71.2 | 84.8	18.8 | 26.5	10.1 | 12.2	1,812.6 | 2,215	33.9 | 57
10-1-1-1	0.1	13.5 | 16.8	111.6 | 127.4	48.1 | 59.9	24.4 | 29.5	1,970.0 | 2,104	50.0 | 50
1-10-1-1	0.0	4.7 | 5.9	39.1 | 44.0	41.0 | 51.7	32.0 | 36.5	1,341.8 | 1,449	37.1 | 44
1-1-10-1	0.1	15.5 | 27.5	115.9 | 147.8	58.9 | 101.6	28.2 | 36.2	2,045.9 | 2816	60.0 | 100
1-1-1-10	0.0	3.8 | 12.1	38.6 | 60.2	23.0 | 56.8	20.2 | 28.7	1,086.8 | 1,979	23.6 | 86

reason is that the number of constraints in the MP is increasing by 26 for every additional day, whereas the number of constraints in the SP is increasing by 11. This effect can be seen by the average values for the MP and SP solution time, i.e., Column 4 and 6. The solution time of the SPs seems to increase linearly with the number of days, i.e., 5.8 ms for 7 days, 10.1 ms for 14 days, and 22.4 ms for 28 days. This property does not exist for the average solution time of the MP, i.e., 11.5 ms for 7 days, 71.2 ms for 14 days, and 512.4 ms for 28 days. This increase in solution time corresponds quite well to the number of generated columns that is given in Column 7. The number of explored nodes in the B&P tree is given in Column 8. Although the algorithm can find the optimal solution for small instances quickly, i.e., on average 12.7 nodes until termination for a 7-day time horizon, the algorithm searches on average more than 123 nodes to find the optimal solution. One reason is that much more branching decisions need to be considered for large instances, i.e., for a 28 days time horizon.

In a subsequent experiment, our aim is to evaluate the effectiveness of our algorithm by adjusting the weights in the objective function. The study uses the same set of 10 instances as in the previous experiment, spanning a time horizon of 14 days. The baseline setting involves assigning equal weights, where $c^{over} = c^{time} = c^{hour} = 1$, and shift preferences set to 8 and 0. Then we systematically vary each weight by a factor of 10. The aggregated results are detailed in Table 6, following a format similar to the previous analysis. The first column of the table outlines the weight configurations in the objective function, denoted as c^{over}-c^{time}-c^{hour}-c^{shift}.

The first row highlights the base case. Instances with a focus on overstaffing and overtime could only be solved in reasonable time by adding a primal heuristic to the algorithm solving the master problem as a binary program ever 50 nodes with all columns generated. For all other instances/settings, the algorithm converged fast without using a primal heuristic. This result highlights the challenge of dealing with overstaffing and overtime. The average time to solve the MP increased by 50% and 73%, respectively. Focusing on deviations in start times of shifts or shift preferences, the performance improves compared to the base case. The average number of columns needed to solve the problem optimal decreases by 25% and 40%, respectively.

The results show that our algorithm performs well in different problem sizes and objective weightings, and optimal solutions can be found in minutes. However, putting a stronger focus on overstaffing and overtime requires the use of a primal heuristic to speed up the solution process. In the following section, we will analyze and discuss a real-world situation of physician scheduling that focuses on a two-week time horizon and adjusting the safety level α.

Table 7
Responsibilities per physician using different break assignment strategies for the real-world case.

Algorithm	Average	Maximum	Minimum	Break assignment viability score
B&P	0.69	1.00	0.17	129.54
FA	0.70	1.18	0.13	89.23
BA	0.67	1.08	0.15	99.63

5.3. Analysis of scheduling anesthesiologists using real-world data

This study focuses on rostering of anesthesiologists for the operating room theater in a German training hospital with more than 1500 beds. The rosters are generated for a time horizon of two weeks, i.e., $|D| = 14$. On each day, one of the physicians has an overnight duty. Based on the different contract types, i.e., part- and full-time, the total working hours are 1440. Analyzing the demand profile provided by the hospital, we can see that at least 1536 h are needed to perform all operations (see Appendix C). This will result in overtime for physicians and might influence the attractiveness of the hospital as an employer. Additionally, hospital management often accepts overtime, as physicians are willing to work overtime, i.e., many contracts have a clause in which the physician agrees to work overtime [49].

In the initial test, we will assess the effectiveness of our break-time windows by comparing the optimal assignment from our model with two common construction algorithms for breaks: a forward assignment (FA) and a backward assignment (BA). Depending on the number of physicians available per shift, we assign the first physician of the shift to the earliest (latest) possible break period, the subsequent physician to the second (second-to-last) possible break period, etc. If there are more physicians than available break periods, we loop back to the first (last) possible break period. This idea tries to spread breaks evenly in their respective break windows. For each period in the schedule, we will record the number of surgeries that an anesthetist is responsible for. The aggregated results are detailed in Table 7.

The key point to note is that FA and BA lead to anesthetists being responsible for multiple patients, as shown in the third column. In general, both construction methods tend to exhibit more variability when comparing the highest and lowest levels of responsibility against the optimal outcome of the B&P. Obviously, our model excels in terms of the objective value linked to the likelihood of breaks in the fifth column (see Objective (1a)), with a reduction of up to 31%. A graphical representation of the responsibilities on a particular day can be seen in Fig. 5.

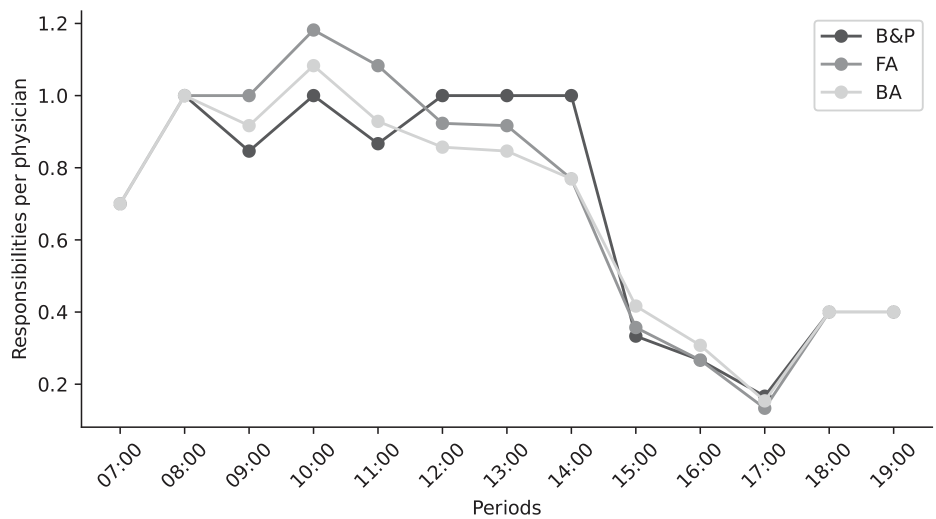

Fig. 5. Responsibilities per anesthetist and period for one specific day.

Table 8
Responsibilities per physician using different break assignment strategies for the Monte Carlo simulation (N=1,000).

Algorithm	Average	Maximum	Minimum
B&P	0.49	1.15	0.00
FA	0.50	1.36	0.00
BA	0.48	1.25	0.00

It is evident that the bottlenecks occur in the morning hours before noon. This is also the contributing factor to FA's lower performance compared to BA. From 12 to 14 o'clock, B&P schedules breaks for all anesthetists who are not needed.

To evaluate the impact of varying durations of a surgery, a Monte Carlo simulation was performed with 1000 iterations. The results are detailed in Table 8. It is evident from the second column that the average responsibility per physician and period remains consistent across all break assignment strategies. Although the minimum responsibility is uniformly 0 in all methods, variations in the maximum value are observed. In particular, our proposed model demonstrates the lowest responsibility at 1.15, compared to 1.36 and 1.25 for FA and BA, respectively. These results suggest that our model outperforms common construction heuristics, which might reflect real-world planning, in optimizing break times and provides a robust solution when surgery durations change.

In a second test, we analyze the effect of different safety level in Constraints (1f) by varying the α-value, i.e., $\alpha \in \{0.6, 0.7, 0.75, 0.8, 0.81, 0.815\}$. Remember, the α-value can be seen as the average day specific safety level for break viability, i.e., for $\alpha = 0.8$ a break can only be given in period p if the average viability score of being able to take a break in period p over all scenarios $n \in N$ is greater than or equal to 0.8. As the performance of the algorithm is evaluated in Section 5.2 only the managerial results are given in Table 9. However, the computational results are stated in Appendix D.

The identification of the instance is given in the first column by labeling the α-value. Column 2 and 3 present the total number of overstaffing, as well as the average overstaffing per period. As expected, the overstaffing increases with the α-value, i.e., 182 periods of overstaffing with $\alpha = 0.6$ in contrast to 266 periods of overstaffing with $\alpha = 0.815$. The overtime hours of all physicians and the average overtime per week are given in columns 4 and 5. Similarly to overstaffing, the value increases with a higher α-value. However, the increase is not as strong as for overstaffing, i.e., overstaffing increases by more than 46% and overtime by 30%. The high overtime is no surprise because we have already discussed that the workforce is not able to handle the complete demand with the given contract types, i.e., 136 periods are not covered. The last two columns present the difference in starting periods of the given shifts. The impact of the α-value is quite high, i.e., the total value is almost tripled. However, the difference in the starting time of a shift is on average less than an hour for an individual physician. One reason is that the roster of most physicians starts on the same period every day of the week. For example, we have 8 physicians starting every day in the same period in the optimal solution with $\alpha = 0.8$. Moreover, only two physicians have a difference in their starting period that is larger than 2 periods. The final schedule with $\alpha = 0.8$ is given in Table 10. The numbers in the matrix indicate the specific shift for each physician and day. As a first result, we can say that considering breaks does influence the generated schedules and affects overstaffing as well as overtime. With this information at hand, the hospital's management is able to define a price for breaks, e.g., increasing the viability of a break by 5% generates 5 overtime hours. However, an important question in this context is how demand should be determined. Given a fixed shift system, it is important that the demand profiles, e.g., surgery planning, consider the shift system of the employees, i.e., nurses and physicians.

As the managerial performance of the generated rosters is not good at all, i.e., high overstaffing and overtime hours, we will next analyze possible opportunities for the hospital management. We will use the solution with α of 0.8 as the base case for the following tests. First, we will evaluate the influence of using alternative contract types. Therefore, we change the regular weekly working hours to 45 for full-time employees and 30 for part-time employees (instance 1). Second, we analyze the effect of not using part-time physicians at all, i.e., all 20 physicians are full-timers (instance 2). Finally, we analyze the effect of using a new 6-hour shift for part-time physicians that will cover periods from 4 to 9 and do not need a break due to working regulations (instance 3), i. e., the new shift is not considered in the set S of Constraints (4c) and (4d) and is only available for part-timers in the set I. An overview of the different input parameters is given in Table 11 and the results are stated in Table 12.

Table 9
Managerial results with parameter variation in α.

Alpha	Overstaffing	Avg. overstaffing per period	Overtime	Avg. overtime per week	Starting time difference	Avg. starting time difference per week
0.6	182	1.0	278	7.0	11	0.3
0.7	182	1.0	278	7.0	11	0.3
0.75	182	1.0	278	7.0	11	0.3
0.8	204	1.1	300	7.5	20	0.5
0.81	238	1.3	334	8.4	32	0.8
0.815	266	1.5	362	9.1	31	0.8
Avg.	209.0	1.1	305.0	7.6	19.3	0.5

Table 10
Final schedule of the base case.

$i\backslash d$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	6[a]	–	1	2	1	–	–	–	1	6	4	4	–	–
2	2	6[a]	–	2	4	–	–	2	2	2	2	2	–	–
3	2	2	6[a]	–	2	–	–	3	3	3	3	3	–	–
4	2	2	4	6[a]	–	–	–	2	2	2	2	2	–	–
5	2	1	2	2	6[a]	–	–	4	–	6	4	1	–	–
6	2	2	1	2	4	[b]	–	4	4	–	4	4	–	–
7	2	1	1	2	4	–	[b]	–	4	4	4	4	–	–
8	1	1	1	1	1	–	–	6[a]	–	1	1	1	–	–
9	3	3	3	3	3	–	–	–	6[a]	–	1	1	–	–
10	3	3	3	3	2	–	–	2	–	6[a]	–	2	–	–
11	1	4	4	4	4	–	–	4	4	4	6[a]	–	–	–
12	3	5	2	3	2	–	–	2	–	5	2	6[a]	–	–
13	2	2	2	2	2	–	–	2	2	2	2	2	[b]	–
14	1	4	4	1	1	–	–	1	1	1	1	1	–	[b]
15	1	1	4	1	1	–	–	3	3	3	3	3	–	–
16	–	4	4	1	4	–	–	–	3	3	3	3	–	–
17	3	–	3	3	3	–	–	2	–	5	2	2	–	–
18	2	2	–	2	2	–	–	2	2	–	2	2	–	–
19	4	4	4	–	4	–	–	4	–	4	–	4	–	–
20	2	2	2	2	–	–	–	2	1	1	1	–	–	–

[a] ≙duty.
[b] ≙weekend duty.

Table 11
Overview of input variation for the case study.

Instance	Full-/part-timer	Working time	Shifts
Base Case	15/5	40/24	6
1	15/5	**45/30**	6
2	**20/0**	40/24	6
3	15/5	40/24	**7**

The information of the instance is given in Column 1. Overstaffing and the average overstaffing per period are given in Column 2 and 3. A surprising result is that only the replacement of part-timer by full-timer (instance 2) gives a lower overstaffing compared to the base case. However, none of the alternative instances can reduce the overstaffing significantly. The total overtime hours and the average overtime per week and the physician are given in Column 4 and 5. As expected, overtime can be reduced by increasing the total weekly working hours from 40 to 45 and 24 to 30 as well as by replacing part-timers with full-time positions. However, generating a new six-hour shift increases overtime hours because full-time positions compensate for the reduction of working hours for part-timers by the new shift. The difference in starting periods of the given shifts is presented in Column 6 and the average difference in starting periods per day in Column 7. For all instances, the value stays almost the same. However, a positive effect can be seen for instances that use different working hours.

From a managerial perspective, the consideration of breaks does influence the generated schedules and affects overstaffing as well as overtime. However, there is no possibility of ignoring them. In the second part of our study, we showed that the reduction of part-timers had a positive effect on overstaffing. Although this result is not valid for any case, hospital management should consider this opportunity as well.

6. Conclusion and future research

Scheduling physicians is a highly complex task, given the multitude of individual requirements involved. Unfortunately, the importance of incorporating breaks into the scheduling process is often overlooked, resulting in critical consequences, such as increased mortality rates due to reduced attention. This paper has introduced a novel model and method aimed at shedding light on this issue and provide an initial solution for real-world applications.

The proposed model formulated as an MIP approach takes into account the viability of breaks employing a new data-driven scoring method. Given the complexity of the formulation, we have decomposed the model and presented a B&P algorithm to effectively solve real-world instances. In particular, by implementing a problem-specific variable selection strategy for branching, the algorithm has proven to be versatile enough to handle different time horizons, ranging from one to four weeks. The performance of the algorithm using different weight combinations highlights the challenges when focusing on overstaffing and overtime.

We have developed a method and outlined its application to define appropriate scores for the viability of break periods using a survival regression model. By understanding the likelihood of break times, hospitals can more effectively plan their staffing needs. This can lead to more strategic scheduling that ensures coverage during peak demand times while still accommodating necessary breaks for staff. It can help avoid understaffing or overstaffing situations, optimizing operational costs. Professionals have the opportunity to utilize this concept and implement it in various contexts, considering changes in input variables that influence the break periods. Moreover, our method can be applied in different planning settings. Although our study focused on the asynchronous planning situation in which the rosters are planned before the task schedule (such as a surgery schedule), the other way around can be easily applied, ignoring the different scenarios $n \in N$ since each task is planned and fixed. The same is true for an integrated planning approach in which rosters and tasks are planned simultaneously.

Furthermore, we have employed various key performance indicators to analyze the impact of considering the viability of breaks, revealing its influence on mitigating both overstaffing and overtime hours in hospitals. Although hospitals can estimate the financial cost of incorporating breaks, the consequences of disregarding them remain crucial.

Table 12
Managerial results of different input settings.

Alpha	Overstaffing	Avg. overstaffing per period	Overtime	Avg. overtime per week	Starting time difference	Avg. starting time difference per week
Base Case	204	1.1	300	7.5	20	0.5
1	218	1.2	104	2.6	19	0.5
2	182	1.0	118	2.95	17	0.4
3	226	1.2	322	8.05	21	0.5

It is essential that hospital management is aware of this problem, as breaks are legally mandated, yet often neglected because of planning challenges. The fixed-shift system poses a particular challenge, emphasizing the need for demand profiles (e.g., surgery planning) that take physicians' shift schedules into account. This approach can effectively reduce overstaffing while enhancing employee satisfaction.

Despite the practical relevance of breaks in the healthcare domain, the topic has not received sufficient attention in the physician literature. This paper has focused on increasing the likelihood that breaks occur during shifts by exploring various scenarios. A potential extension to our formulation could involve considering a minimum likelihood of breaks in each scenario, providing a different perspective for analyzing the risks of not taking breaks. Additionally, it is crucial to consider other aspects related to breaks. For instance, the impact of impractical breaks should not be limited to individual physicians; fairness, a commonly used metric for assessing ergonomic effects, could serve as a useful tool for evaluating breaks' effectiveness as well. Such comprehensive considerations can further improve physician scheduling practices and ultimately improve overall healthcare outcomes. Integration of our concept into stochastic formulations could lead to a better estimate of costs and a more realistic staffing. However, a better evaluation of the impacts of integrating demand uncertainty in combination with the viability score of break-time windows is necessary. One way to address this problem is to simultaneously solve the surgery scheduling problem with the physician scheduling problem. Although there is no straightforward way to align the deadline and planning horizons of the different approaches, other areas of application might be able to handle this situation more effectively.

Funding

This research project is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) and Grant no. 438507036.

CRediT authorship contribution statement

Sebastian Kraul: Writing – review & editing, Writing – original draft, Visualization, Software, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Melanie Erhard:** Writing – review & editing, Validation, Resources, Methodology, Investigation, Conceptualization. **Jens O. Brunner:** Writing – review & editing, Validation, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jens O. Brunner reports financial support was provided by the German Research Foundation (Grant no. 438507036).

Data availability

Data will be made available on request.

Declaration of generative AI in scientific writing

During the preparation of this work, the author(s) used ChatGPT and Grammarly to improve readability and language. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Appendix A. Construction of the scenarios *N*

Algorithm 1 Generate o^n_{spd} for all scenarios

Require: $P, D, S, N, A_{sp}, F_s, L_s, H_s, B^{pre}_s, B^{post}_s$
Ensure: o^n_{spd} $\forall s \in S, p \in P, d \in D, n \in N$
  for $n \in N$ **do**
    for $s \in S$ **do**
      for $d \in D$ **do**
        for $p \in P$ **do**
          $o^n_{spd} \leftarrow 0$
          if p in *break window* of shift s **then**
            o^n_{spd} = random.uniform(0.6; 1)
          end if
        end for
      end for
    end for
  end for

Appendix B. AFT model parameters

See Table 13.

Appendix C. Data of the test instances

See Tables 14–17.

Appendix D. Computational results

See Table 18.

Table 13
Log–logistic AFT model coefficients for the most common surgery.

	coef	exp(coef)	se(coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(coef) upper 95%	cmp to	z	p	-log2(p)
Intercept	−2.05	0.13	0.62	−3.27	−0.83	0.04	0.44	0.00	−3.28	<0.005	9.93
Hour	−0.07	0.94	0.89	−1.80	1.67	0.17	5.32	0.00	−0.07	0.94	0.09
Rolling_avg	5.02	150.81	1.32	2.43	7.60	11.38	1998.66	0.00	3.80	<0.005	12.78
Age	−0.28	0.76	0.72	−1.69	1.14	0.18	3.13	0.00	−0.38	0.70	0.51
Secondary	0.01	1.01	0.89	−1.74	1.75	0.18	5.78	0.00	0.01	0.99	0.01
Rolling_avg_prev	−0.31	0.73	1.31	−2.88	2.26	0.06	9.62	0.00	−0.24	0.81	0.30
ICU	−0.03	0.97	0.04	−0.10	0.04	0.91	1.05	0.00	−0.74	0.46	1.12
Hour:Rolling_avg	0.23	1.26	1.97	−3.63	4.10	0.03	60.29	0.00	0.12	0.91	0.14
Rolling_avg:Age	0.94	2.56	1.64	−2.27	4.15	0.10	63.46	0.00	0.57	0.57	0.82
Hour:Secondary	0.08	1.09	1.86	−3.56	3.73	0.03	41.55	0.00	0.05	0.96	0.05
Rolling_avg:Secondary	0.97	2.64	1.97	−2.89	4.83	0.06	124.74	0.00	0.49	0.62	0.68
Age:Secondary	−0.43	0.65	1.27	−2.93	2.06	0.05	7.86	0.00	−0.34	0.73	0.45
Hour:Rolling_avg_prev	−0.36	0.70	1.92	−4.12	3.40	0.02	29.92	0.00	−0.19	0.85	0.23
Rolling_avg:Rolling_avg_prev	−3.81	0.02	2.20	−8.12	0.49	0.00	1.64	0.00	−1.74	0.08	3.60
Age:Rolling_avg_prev	−0.12	0.89	1.62	−3.29	3.05	0.04	21.19	0.00	−0.07	0.94	0.09
Secondary:Rolling_avg_prev	−0.53	0.59	1.98	−4.41	3.35	0.01	28.46	0.00	−0.27	0.79	0.34
Hour:Rolling_avg:Age	0.53	1.70	2.75	−4.86	5.92	0.01	373.88	0.00	0.19	0.85	0.24
Hour:Rolling_avg:Secondary	1.43	4.19	4.16	−6.72	9.58	0.00	14 504.69	0.00	0.34	0.73	0.45
Hour:Age:Secondary	0.07	1.07	2.67	−5.16	5.30	0.01	200.42	0.00	0.03	0.98	0.03
Rolling_avg:Age:Secondary	0.48	1.62	2.84	−5.07	6.04	0.01	420.82	0.00	0.17	0.86	0.21
Hour:Rolling_avg:Rolling_avg_prev	−0.01	0.99	3.16	−6.20	6.17	0.00	480.27	0.00	−0.00	1.00	0.01
Hour:Age:Rolling_avg_prev	0.05	1.05	2.72	−5.29	5.38	0.01	217.33	0.00	0.02	0.99	0.02
Rolling_avg:Age:Rolling_avg_prev	−0.96	0.38	2.66	−6.18	4.26	0.00	70.52	0.00	−0.36	0.72	0.48
Hour:Secondary:Rolling_avg_prev	−2.03	0.13	4.22	−10.29	6.24	0.00	512.23	0.00	−0.48	0.63	0.66
Rolling_avg:Secondary:Rolling_avg_prev	0.07	1.07	3.29	−6.38	6.53	0.00	682.60	0.00	0.02	0.98	0.03
Age:Secondary:Rolling_avg_prev	−0.96	0.38	2.92	−6.68	4.76	0.00	116.31	0.00	−0.33	0.74	0.43
Hour:Rolling_avg:Age:Secondary	0.02	1.02	6.07	−11.88	11.92	0.00	1.51e+05	0.00	0.00	1.00	0.00
Hour:Rolling_avg:Age:Rolling_avg_prev	0.02	1.02	4.62	−9.04	9.07	0.00	8699.72	0.00	0.00	1.00	0.00
Hour:Rolling_avg:Secondary:Rolling_avg_prev	0.02	1.02	7.17	−14.04	14.08	0.00	1.31e+06	0.00	0.00	1.00	0.00
Hour:Age:Secondary:Rolling_avg_prev	−0.27	0.76	6.20	−12.42	11.88	0.00	1.45e+05	0.00	−0.04	0.97	0.05
Rolling_avg:Age:Secondary:Rolling_avg_prev	2.47	11.83	5.00	−7.33	12.27	0.00	2.14e+05	0.00	0.49	0.62	0.69
Hour:Rolling_avg:Age:Secondary:Rolling_avg_prev	−0.03	0.97	10.52	−20.66	20.59	0.00	8.77e+08	0.00	−0.00	1.00	0.00
Intercept	2.90	18.26	0.02	2.86	2.95	17.41	19.14	0.00	120.07	<0.005	inf

Table 14
Demand profile for the test instances with 7, 14, 21, and 28 days.

$d \setminus p$	1	2	3	4	5	6	7	8	9	10	11	12	13
1	6	7	11	12	14	14	14	14	14	14	6	5	5
2	9	11	13	14	14	14	14	14	14	14	5	4	3
3	11	11	11	12	14	14	14	14	14	14	3	2	1
4	6	7	11	12	14	14	14	14	14	14	6	5	5
5	11	11	11	12	14	14	14	14	14	14	3	2	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	5	9	10	13	8	13	13	13	13	13	0	0	0
9	7	10	10	11	11	11	11	11	11	11	2	1	0
10	8	12	12	14	14	14	14	14	14	14	9	9	8
11	9	11	11	14	14	14	14	14	14	14	2	0	0
12	9	11	13	13	14	14	14	14	14	14	1	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0
15	9	9	12	13	13	13	13	12	12	11	4	2	2
16	5	7	13	13	13	13	11	11	11	11	4	3	1
17	5	12	13	13	13	12	11	11	11	11	7	5	2
18	8	9	12	12	13	11	11	11	11	11	9	5	5
19	10	12	12	13	13	13	13	13	12	12	2	2	2
20	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0
22	8	12	12	13	13	13	12	11	11	11	7	2	2
23	10	10	10	13	14	13	11	11	11	11	9	0	0
24	9	9	10	13	13	13	12	12	12	11	1	1	1
25	10	12	13	13	13	13	12	12	12	12	5	5	5
26	7	8	10	12	12	12	11	11	11	11	1	1	1
27	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15
Duty schedule (K_{id}) for the test instances with 7, 14, 21, and 28 days.

$d\backslash i$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
16	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Table 16
Availability of physicians (V_{id}) for the test instances with 7, 14, 21, and 28 days.

$d\backslash i$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
3	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
4	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
5	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1
9	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1
10	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1
11	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1
12	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1
17	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
18	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
19	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1
23	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1
24	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1
25	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1
26	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 17
Shift matrix A_{sp} for the test instances with 7, 14, 21, and 28 days.

$s \backslash p$	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	1	1	1	1	1	1	1	1				
2		1	1	1	1	1	1	1	1	1			
3					1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1		
5			1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 18
Computational results with parameter variation in α.

Alpha	LB	UB	LB Root	GAP (%)	MP Time (s)	Avg. MP Time (ms)	SP Time (s)	Avg. SP Time (ms)	No. columns gen	No. nodes expl
0.6	−1,811.7	−1,811.6	−1,811.7	0.0	14.8	104.2	29.0	13.1	2,211	58
0.7	−1,811.7	−1,811.6	−1,811.7	0.0	31.2	120.9	54.3	16.5	3,286	107
0.75	−1,811.7	−1,811.7	−1,811.7	0.0	39.4	113.2	79.2	16.4	4,815	168
0.8	−1,780.9	−1,780.7	−1,780.9	0.0	14.4	84.7	44.4	20.2	2,196	69
0.81	−1,732.9	−1,732.4	−1,732.9	0.0	13.1	62.4	55.2	24.3	2,271	69
0.815	−1,687.1	−1,687.1	−1,687.1	0.0	24.6	87.9	102.9	31.7	3,243	225
Avg.	−1,772.7	−1,772.5	−1,772.7	0.0	22.9	95.6	60.8	20.4	3,003.7	116.0

References

[1] Statistisches Bundesamt. Gesundheit: Grunddaten der Krankenhäuser 2016. Fachserie 2016;12(6):1–156.

[2] Schelhase T. Statistische Krankenhausdaten: Diagnosedaten der Krankenhäuser 2011. In: Klauber J, Beivers A, Augurzky B, editors. Krankenhaus-Report 2014. Stuttgart: Schattauer; 2014, p. 329–56.

[3] Bölt U, Graf T. Gesundheit-20 Jahre Krankenhausstatistik. Wirtschaft Stat 2012;(2):112–39.

[4] Blum K, Löffert S, Offermanns M, Steffen P. Krankenhaus Barometer: Umfrage 2013. 2013. URL https://www.dki.de/fileadmin/publikationen/krankenhaus_barometer_2013_1_.pdf.

[5] Schimmelpfeng K, Helber S, Kasper S. Decision support for rehabilitation hospital scheduling. OR Spectrum 2012;34(2):461–89. http://dx.doi.org/10.1007/s00291-011-0273-0.

[6] Kraul S. Annual scheduling for anesthesiology medicine residents in task-related programs with a focus on continuity of care. Flex Serv Manuf J 2020;32(1):181–212. http://dx.doi.org/10.1007/s10696-019-09365-4.

[7] Fügener A, Brunner JO, Podtschaske A. Duty and workstation rostering considering preferences and fairness: A case study at a department of anaesthesiology. Int J Prod Res 2015;53(24):7465–87. http://dx.doi.org/10.1080/00207543.2015.1082667.

[8] Silva TAO, de Souza MC. Surgical scheduling under uncertainty by approximate dynamic programming. Omega 2020;95:102066. http://dx.doi.org/10.1016/j.omega.2019.05.002.

[9] Damcı-Kurt P, Zhang M, Marentay B, Govind N. Improving physician schedules by leveraging equalization: Cases from hospitals in U.S. Omega 2019;85:182–93. http://dx.doi.org/10.1016/j.omega.2018.06.011.

[10] Erhard M, Schoenfelder J, Fügener A, Brunner JO. State of the art in physician scheduling. Eur J Oper Res 2018;265(1):1–18. http://dx.doi.org/10.1016/j.ejor.2017.06.037.

[11] Baum R, Bertsimas D, Kallus N. Scheduling, revenue management, and fairness in an academic-hospital radiology division. Academic Radiol 2014;21(10):1322–30. http://dx.doi.org/10.1016/j.acra.2014.05.009.

[12] Shamia O, Aboushaqrah N, Bayoumy N. Physician on call scheduling: Case of a Qatari hospital. In: 2015 6th international conference on modeling, simulation and applied optimization. Piscataway, NJ: IEEE; 2015, p. 1–6. http://dx.doi.org/10.1109/ICMSAO.2015.7152217.

[13] Bowers MR, Noon CE, Wu W, Bass JK. Neonatal physician scheduling at the university of Tennessee medical center. Interfaces 2016;46(2):168–82. http://dx.doi.org/10.1287/inte.2015.0839.

[14] Marchesi JF, Hamacher S, Fleck JL. A stochastic programming approach to the physician staffing and scheduling problem. Comput Ind Eng 2020;106281. http://dx.doi.org/10.1016/j.cie.2020.106281.

[15] Tohidi M, Kazemi Zanjani M, Contreras I. A physician planning framework for polyclinics under uncertainty. Omega 2021;101:102275. http://dx.doi.org/10.1016/j.omega.2020.102275.

[16] Wang F, Zhang C, Zhang H, Xu L. Short-term physician rescheduling model with feature-driven demand for mental disorders outpatients. Omega 2021;105:102519. http://dx.doi.org/10.1016/j.omega.2021.102519.

[17] Brunner JO, Bard JF, Kolisch R. Flexible shift scheduling of physicians. Health Care Manag Sci 2009;12(3):285–305. http://dx.doi.org/10.1007/s10729-008-9095-2.

[18] Brunner JO, Bard JF, Kolisch R. Midterm scheduling of physicians with flexible shifts using branch and price. IIE Trans 2010;43(2):84–109. http://dx.doi.org/10.1080/0740817X.2010.504685.

[19] Erhard M. Flexible staffing of physicians with column generation. Flex Serv Manuf J 2019;9(2):114. http://dx.doi.org/10.1007/s10696-019-09353-8.

[20] Stolletz R, Brunner JO. Fair optimization of fortnightly physician schedules with flexible shifts. Eur J Oper Res 2012;219(3):622–9. http://dx.doi.org/10.1016/j.ejor.2011.10.038.

[21] van den Bergh J, Beliën J, de Bruecker P, Demeulemeester E, de Boeck L. Personnel scheduling: A literature review. Eur J Oper Res 2013;226(3):367–85. http://dx.doi.org/10.1016/j.ejor.2012.11.029.

[22] Bard JF, Morton DP, Wang YM. Workforce planning at USps mail processing and distribution centers using stochastic optimization. Ann Oper Res 2007;155(1):51–78. http://dx.doi.org/10.1007/s10479-007-0213-1.

[23] Bechtold SE, Jacobs LW. Implicit modeling of flexible break assignments in optimal shift scheduling. Manag Sci 1990;36(11):1339–51. http://dx.doi.org/10.1287/mnsc.36.11.1339.

[24] Aykin T. Optimal shift scheduling with multiple break windows. Manag Sci 1996;42(4):591–602. http://dx.doi.org/10.1287/mnsc.42.4.591.

[25] Sungur B, Özgüven C, Kariper Y. Shift scheduling with break windows, ideal break periods, and ideal waiting times. Flex Serv Manuf J 2017;29(2):203–22. http://dx.doi.org/10.1007/s10696-015-9234-2.

[26] Kiermaier F, Frey M, Bard JF. The flexible break assignment problem for large tour scheduling problems with an application to airport ground handlers. J Sched 2020;23(2):177–209. http://dx.doi.org/10.1007/s10951-019-00635-5.

[27] Cohn A, Root S, Kymissis C, Esses J, Westmoreland N. Scheduling medical residents at Boston university school of medicine. Interfaces 2009;39(3):186–95. http://dx.doi.org/10.1287/inte.1080.0369.

[28] Elomri A, Elthlatiny S, Sidi Mohamed Z. A goal programming model for fairly scheduling medicine residents. Int J Supply Chain Manag 2015;4(2).

[29] Proano RA, Agarwal A. Scheduling internal medicine resident rotations to ensure fairness and facilitate continuity of care. Health Care Manag Sci 2018;21(4):461–74. http://dx.doi.org/10.1007/s10729-017-9403-9.

[30] Hong Y-C, Cohn A, Gorga S, O'Brien E, Pozehl W, Zank J. Using optimization techniques and multidisciplinary collaboration to solve a challenging real-world residency scheduling problem. Interfaces 2019. http://dx.doi.org/10.1287/inte.2019.0995.

[31] Ernst A, Jiang H, Krishnamoorthy M, Sier D. Staff scheduling and rostering: A review of applications, methods and models. Eur J Oper Res 2004;153(1):3–27. http://dx.doi.org/10.1016/S0377-2217(03)00095-X.

[32] Bard JF, Purnomo HW. Preference scheduling for nurses using column generation. Eur J Oper Res 2005;164(2):510–34. http://dx.doi.org/10.1016/j.ejor.2003.06.046.

[33] Gross CN, Brunner JO, Blobner M. Hospital physicians can't get no long-term satisfaction - an indicator for fairness in preference fulfillment on duty schedules. Health Care Manag Sci 2018. http://dx.doi.org/10.1007/s10729-018-9452-8.

[34] Adams T, O'Sullivan M, Walker C. Physician rostering for workload balance. Oper Res Health Care 2019;20:1–10. http://dx.doi.org/10.1016/j.orhc.2018.11.001.

[35] Rea D, Froehle C, Masterson S, Stettler B, Fermann G, Pancioli A. Unequal but fair: Incorporating distributive justice in operational allocation models. Prod Oper Manag 2021;30(7):2304–20. http://dx.doi.org/10.1111/poms.13369.

[36] Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH. Branch-and-price: Column generation for solving huge integer programs. Oper Res 1998;46(3):316–29. http://dx.doi.org/10.1287/opre.46.3.316.

[37] Dantzig GB, Wolfe P. Decomposition principle for linear programs. Oper Res 1960;8(1):101–11. http://dx.doi.org/10.1287/opre.8.1.101.

[38] Desrosiers J, Lübbecke ME. A primer in column generation. In: Desaulniers G, editor. Column generation. GERAD 25th anniversary series, New York, NY: Springer; 2005, p. 1–32. http://dx.doi.org/10.1007/0-387-25486-2_1.

[39] Vanderbeck F. On dantzig-wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. Oper Res 2000;48(1):111–28. http://dx.doi.org/10.1287/opre.48.1.111.12453.

[40] Desrosiers J, Lübbecke ME. Branch-price-and-cut algorithms. In: Cochran JJ, editor. Wiley encyclopedia of operations research and management science. Hoboken, NJ: Wiley Interscience; 2011. http://dx.doi.org/10.1002/9780470400531.eorms0118.

[41] Ryan DM, Foster BA. An integer programming approach to scheduling. In: Wern A, editor. Computer scheduling of public transport. North Holland; 1981, p. 269–80.

[42] Vanderbeck F, Wolsey LA. An exact algorithm for IP column generation. Oper Res Lett 1996;19(4):151–9. http://dx.doi.org/10.1016/0167-6377(96)00033-8.

[43] George B, Seals S, Aban I. Survival analysis and regression models. J Nucl Cardiol Off Publ Am Soc Nucl Cardiol 2014;21(4):686–94. http://dx.doi.org/10.1007/s12350-014-9908-2.

[44] Martinussen T, Scheike TH. Dynamic regression models for survival data. In: Statistics for biology and health, New York, NY: Springer; 2006. http://dx.doi.org/10.1007/0-387-33960-4.

[45] Strum DP, May JH, Sampson AR, Vargas LG, Spangler WE. Estimating times of surgeries with two component procedures: comparison of the lognormal and normal models. Anesthesiology 2003;98(1):232–40. http://dx.doi.org/10.1097/00000542-200301000-00035.

[46] Kayis E, Wang H, Patel M, Gonzalez T, Jain S, Ramamurthi RJ, et al. Improving prediction of surgery duration using operational and temporal factors. AMIA Annu Symp Proc 2012;2012:456–62.

[47] Dexter F, Dexter EU, Masursky D, Nussmeier NA. Systematic review of general thoracic surgery articles to identify predictors of operating room case durations. Anesth Analg 2008;106(4):1232–41. http://dx.doi.org/10.1213/ane.0b013e318164f0d5, table of contents.

[48] Ingels J, Maenhout B. Employee substitutability as a tool to improve the robustness in personnel scheduling. OR Spectrum 2017;39(3):623–58. http://dx.doi.org/10.1007/s00291-017-0476-0.

[49] Tarifgemeinschaft deutscher Länder, Marburger Bund. Tarifvertrag für Ärztinnen und Ärzte an Universitätskliniken. Berlin; 2017. URL https://www.marburger-bund.de/sites/default/files/tarifvertraege/2018-08/17-08-29-tv-aerzte-i-d-f-6-aenderungstv.pdf.