Journal Pre-proof

Stable Annual Scheduling of Medical Residents Using Prioritized Multiple Training Schedules to Combat Operational Uncertainty

Sebastian Kraul, Jens O. Brunner

PII: S0377-2217(23)00123-6
DOI: https://doi.org/10.1016/j.ejor.2023.02.007
Reference: EOR 18341

To appear in: *European Journal of Operational Research*

Received date: 31 January 2022
Accepted date: 7 February 2023

Please cite this article as: Sebastian Kraul, Jens O. Brunner, Stable Annual Scheduling of Medical Residents Using Prioritized Multiple Training Schedules to Combat Operational Uncertainty, *European Journal of Operational Research* (2023), doi: https://doi.org/10.1016/j.ejor.2023.02.007

Highlights

- We analyze the mismatch between annual training scheduling and daily rostering.
- We use a two-stage formulation to appropriately represent the planning processes and develop an analytic bound for our formulation.
- We solve our problem with a sample average approximation using a decomposition strategy.
- We use real-world data in anesthesiology provided by a German training hospital to evaluate our approach.

Stable Annual Scheduling of Medical Residents Using Prioritized Multiple Training Schedules to Combat Operational Uncertainty

Sebastian Kraul[a*] · Jens O. Brunner[b,c]

[a]Department of Operations Analytics, Vrije Universiteit Amsterdam, Amsterdam, the Netherlands
[b]Chair of Health Care Operations/ Health Information Management, University of Augsburg, Augsburg, Germany

[c]Department of Technology, Management, and Economics, Technical University of Denmark, Denmark

For educational purposes, medical residents often have to pass through many departments, which place different requirements on them. They are informed about the upcoming departments by an annual training schedule which keeps the individual departments' service level as constant as possible. Due to poor planning and uncertain events, deviations in the schedule can occur. These deviations affect the service level in the departments, as well as the training progress and satisfaction of the residents. This article analyzes the impact of priorities on residents' annual planning based on department assignments to combat uncertainty that might result in departmental changes. We present a novel two-stage formulation that combines residents' tactical planning with duty and daily scheduling's operational level. We determine an analytical bound for the problem that is superior to the LP bound. Additionally, we approximate a bound based on the solution approach using the objective value of the deterministic solution of an instance and the absences in each scenario. In a computational study, we analyze the performance of various bounds, our solution approach, and the effects of additional priorities in residents' annual planning. We show that additional priorities can significantly reduce the number of unexpected department assignments. Finally, we derive a practical number of priorities from the results.

Keywords OR in Health Services · Resident Scheduling · Training Priorities · Two-stage Approach · Integer Programming

Funding This research project is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) and Grant no. 405488489.

*Corresponding Author: s.kraul@vu.nl

Stable Annual Scheduling of Medical Residents Using Prioritized Multiple Training Schedules to Combat Operational Uncertainty

For educational purposes, medical residents often have to pass through many departments, which place different requirements on them. They are informed about the upcoming departments by an annual training schedule which keeps the individual departments' service level as constant as possible. Due to poor planning and uncertain events, deviations in the schedule can occur. These deviations affect the service level in the departments, as well as the training progress and satisfaction of the residents. This article analyzes the impact of priorities on residents' annual planning based on department assignments to combat uncertainty that might result in departmental changes. We present a novel two-stage formulation that combines residents' tactical planning with duty and daily scheduling's operational level. We determine an analytical bound for the problem that is superior to the LP bound. Additionally, we approximate a bound based on the solution approach using the objective value of the deterministic solution of an instance and the absences in each scenario. In a computational study, we analyze the performance of various bounds, our solution approach, and the effects of additional priorities in residents' annual planning. We show that additional priorities can significantly reduce the number of unexpected department assignments. Finally, we derive a practical number of priorities from the results.

Keywords OR in Health Services · Resident Scheduling · Training Priorities · Two-stage Approach · Integer Programming

1 Introduction

All newly graduated physicians must continue their studies in a three to seven years clinical training in a specialist discipline such as anesthesia or orthopedics. In this phase of training, they are called *residents*. Although each specialization has different guidelines, the central concept of each *resident program* consists of rotating through different medical departments (see **??**). Within these rotations, the resident is supposed to observe and learn different types of interventions. The hospital management is obliged to ensure the practicability of the resident program within the given time. For this purpose, the management has different instruments available at different planning levels, i. e., staffing and rostering decisions (**?**). While the instruments are generally similar for all use cases, it still makes sense to look at a specific case.

We will motivate the problem from the case of a German (task-based) resident program (**??**). A characteristic of German programs is that the hospital directly employs the residents, i. e., hospitals have a special interest in taking care of the residents' needs. In this context, an annual *training schedule* considers the residents' training objectives and assigns them to specific departments according to demand requirements on a tactical level. The training schedule covers very often one year (**???**). However, a training schedule not necessarily follows a calendar year. The training schedules are intended to ensure planning reliability and have a direct influence on the daily roster, which is often carried out monthly (**???**). We define the *daily roster* as a combination of two schedules as described below. First, the *duty schedules* allocate residents to overnight and weekend duties and ensure a minimum level of service 24/7, i. e., some residents need to stay in the hospital after regular working hours. Second, the *daily schedules* allocate residents to departments for their regular working time, i. e., from Monday to Friday from 7 to 16 o'clock with a full-time contract. Typically, each department can be associated with a specialty discipline. While the duty schedule specifies when a resident works, the daily schedule specifies where a resident works. Note that the duty schedule affects the daily schedule, i. e., residents cannot be assigned to the daily schedule after an overnight duty (**?**).

Although the training schedule serves only as an input for the operational level, the schedule provides the residents and planners with information transparency and a certain degree of planning reliability (**?**). However, a problem occurs if the daily schedule does not comply with the assignments in the training schedule. For instance, in our case study's real-world data for a whole year, about one-fifth of all daily assignments deviate from the training schedule. As a consequence, the resident has to work in a department for which they could not adequately prepare. Insufficient preparation time has a direct influence on the quality of training as well as the quality of care (**??**). Moreover, the training progress is jeopardized because it cannot be guaranteed to what extent interventions relevant to training are carried out in the deviating department. In Germany, these differences led in some cases to delays within the resident program of up to 50% of the total duration (**?**). This led to a tightening of the requirements for hospitals with regard to resident programs, i. e., hospitals must adhere to the time limits of resident programs and provide the residents with an objectively generated schedule (**?**). Similar situations do exist in other countries as well. In the following, we call a training schedule to be *stable* if all assignments in the daily schedule match the assignments in the training schedule.

The purpose of the paper is the development of a new method to tackle the problem of deviations in the daily schedule from the assignments of the training schedule. As already mentioned, a resident cannot be assigned to the daily schedule on a specific day if they had an overnight duty on the previous day. Depending on the department's demand pattern, another resident may have to be assigned to the respective department to ensure demand coverage. Consequently, a resident is assigned to a department different from the one specified in the training schedule. Besides, residents may be absent, e. g., due to workshops, conferences or illness. In this case, a resident from another department would be asked to cover the absent resident. One approach to take these types of changes into account in the training schedule and how it is partly applied in practice is by setting priorities for the assignments, i. e., a resident is assigned to more departments with different priorities in the training schedule. This gives residents additional information about possible fields of activity and ensures that they are not surprised by short-term changes. Therefore, the reliability of planning is increased. It is important to note that we do not mean preferences by priorities which might be specified by the residents, as is often used synonymously in the literature.

We define *priorities* as the importance of an assignment from the training schedule to the daily schedule. For instance, let a resident be assigned with priority 1 to department ENT and with priority 2 to department MJF in the training schedule for one specific week. Usually, the resident will be assigned every day of this week regular working to department ENT in the daily schedule. However, if there is a shortage in department MJF for one or more days, i. e., due to absences, this resident might be assigned to department MJF on the day with the shortage. While a typical annual training schedule, as discussed in the literature, has a single department assignment per period and resident, a training schedule with, for instance, three priorities also has three potential department assignments per period and resident (see Table 1). However, at the operational level (daily schedule) only one of the potential assignments is applied. In other words, one can think of having more than one training schedule for each resident individually with decreasing importance for assignments in the daily schedule. If a resident cannot be assigned to the department of the first training schedule in the daily schedule in a specific period, the assignment of the second schedule is applied, and so on, i. e., the first training schedule (priority of 1) has the highest valuation.

Table 1: Annual training schedule as discussed in literature (left) and in this paper (right)

Resident	Period 1	Period 2
Jane Doe	ENT	ENT

Resident	Priority	Period 1	Period 2
	1	ENT	ENT
Jane Doe	2	MJF	MJF
	3	ICU	ICU

This kind of training schedule design is not considered in current literature as will be shown in the next section. The contribution of our paper is manifold. First, we present a novel mathematical formulation of the annual resident scheduling problem using priorities. We assign residents with different priorities to departments and maximize the time spent in departments with priorities based on the training goals considering the operational restrictions of duty and daily scheduling. Second, we use a two-stage approach combined with a sample average approximation (SAA) to model the uncertainty of absences of residents over a year and determine stable annual training schedules. We model absences in a broader sense. Third, we present a decomposition to solve the formulation to near-optimality in an iterative process which generates a feasible solution in each iteration. Fourth, we determine an analytic bound for the problem that is superior to the LP bound as well as an approximative bound based on numerical results. Eventually, we analyze our formulation by a real-world case of the resident program in anesthesiology of a German teaching hospital with more than 1,200 beds and 80 residents in the anesthesia program. We identify the minimum number of priorities needed to fulfill full information transparency, i. e., a stable training schedule.

The reminder of this article is organized as follows. In Section 2 we review the state of the art in resident scheduling. The problem formulation as well as the mathematical model is described in Section 3. The SAA and the solution algorithm is discussed in Section 4. The computational study and the analysis of training priorities is performed in Section 5. The paper finishes with a conclusion and discussion in Section 6, along with future research avenues.

2 Related work on resident scheduling

Personnel scheduling problems have been studied in great detail over the last decades. The topics cover numerous application areas such as nurse rostering, call center scheduling, and airline scheduling (**?**). In the last two decades, a new type of personnel scheduling problem was analyzed as a side-area of physician scheduling problems, namely resident scheduling problems. Note that every resident is a physician but not every physician is a resident. One of the main differences of this type of problem is that in addition to the classic shift planning in its various variations (**?**), training goals must be considered, i. e., residents must spend a certain amount of time in several departments (**??**). Note that most of the literature deals with the resident scheduling problem when training requirements are considered. However, there are also some papers dealing with training requirements in other areas as well (**????**).

This review focuses mainly on resident and physician scheduling. An extensive overview of physician scheduling can be found in **?**. **?** analyze the trade-off between scheduling quality and scheduling stability during re-planning of physicians taking into account the simultaneous planning of the duty and daily schedule in a time horizon of one month. Residents are taken into account by using a long-term schedule serving as input for the daily roster. This long-term schedule reflects the training goals and ensures allocation to the preferred departments. Their extensive study shows that the simultaneous planning of the duty and daily schedule considers training aspects by far more than in the sequential planning, i. e., first generating a duty schedule and then a daily schedule. In contrast to this work, we are not interested in actual operational execution, but in generating an annual schedule for residents using operational control mechanisms, i. e., we consider a tactical problem.

Most of the resident scheduling literature deals with annual planning problems. **?** assign residents to different training activities over one year. They minimize the number of irregular assignments, assuming a resident can perform only one activity in a period and has to repeat the activity for a given time before the activity can change. They formulate the problem as a binary problem and solve it using a branch-and-price algorithm after a DantzigWolfe reformulation. **?** construct annual block schedules for family medicine residents in a teaching hospital in the US. They formulate two mixed integer problems (MIP). In the first model, they determine an aggregated rotation schedule per month, taking into account duty assignments as well as fairness aspects in terms of treated patients. In the second formulation, they determine the individual rotation assignments based on the solution of the aggregated schedule. In their case study, they were able to show that their formulation is superior to current planning, which is carried out manually. With their formulation, it was possible to create the block schedules entirely at the beginning of the academic year. **?** design an annual training schedule as well as monthly duty rosters. They focus on balancing the training progress among residents of the same category and year. They solve their model with a standard solver and evaluate the impact of the different objectives on the solution time. A network-based formulation of the resident scheduling problem is presented by **?**. They assign residents weekly to rotations at different hospitals over the year and minimize the fixed and variable cost resulting from assigning residents to hospitals and rotations. They show that the network-based formulation is superior to a MIP model. In a case study, they find out that the American University of the Caribbean could have saved, on average, 19% of their costs per year using the new formulation. **?** also deal with the assignment of residents to different disciplines and hospitals. In contrast to **?**, they do not only consider cost but also the preferences of the residents. They develop different heuristics incorporated in a decomposition yielding near-optimal solutions in a short timespan, even for large real-world instances. **?** build on the previous work and develop a branch-and-price algorithm to solve the problem optimal. They test different branching and pricing strategies to speed up the solution process. **?** as well as **?** focus on fairness aspects in the annual scheduling of residents on a weekly basis. Both papers consider continuity of care, i. e., unnecessary changes between departments/rotations are considered. However, **?** simultaneously solve vacation and rotation planning by maximizing the overall resident satisfaction due to their vacation assignments. In contrast to that, **?** focuses on the rotation planning and minimizes the number of departmental changes per week as well as the maximum deviation from the predefined training goals between the residents. One advantage of both papers is that not only one schedule is generated but several. This allows the hospital management to choose from a number of schedules with different strengths and weaknesses. **?** assign a heterogeneous group of medical students to different rotations over a six-month period. They use a variable neighborhood search to evaluate linear and non-linear objectives which measure the stability of the allocations. **?** assign medical students to internships over one year. They deal with the requirements of different stakeholders like the university, the hospital, and the students. They decompose their formulation by time and by students. They combine the Hungarian method with a dynamic programming algorithm in a step-wise optimization algorithm to solve their entire problem. All the considered papers generate an (annual) training schedule for the residents by assigning them to exactly one department in each period. However, our approach generates an annual training schedule that includes more than one possible department assignment per period.

The smaller the time horizon of the problem under consideration, the more detailed the models become. **?** and **?** construct shift schedules on a monthly basis for residents taking into account different types of preferences. The MIP formulation of **?** considers 12 different types of preferences, which are defined not only by residents but also by hospital management. They use various pre- and post-processing steps to solve their formulation in a hierarchical order as a goal programming approach. In contrast, we use the operational level as a tool for evaluating our annual training schedule, i. e., to ensure the stability of the training schedule. We adopt the planning mechanisms of the operational level but do not use individual preferences of residents in the modeling.

The feasibility of training is often analyzed in light of problems related to staffing decisions (**??**). **?** minimize the tardiness of training in a surgical department. In a MIP formulation, they determine the number and type of surgical interventions a resident has to perform in a month based on the hospital's portfolio. To solve real-world instances, they develop a decomposition approach that they combine with a local search technique. The hospital's portfolio of interventions is also considered by **?**. They determine the total number of residents that can finish their training in time, taking into account an uncertain number of interventions relevant for training per period. They propose a robust MIP formulation and use a decomposition heuristic to solve the problem near-optimal in a short time. Their experimental study analyzes the effect of conservatism concerning the uncertain interventions and the resulting number of training positions. Apart from this work, only **?** consider an uncertain environment for resident scheduling problems. They develop a simulation to test the operational practicability of different resident schedules. Their analysis's primary focus is the total and the maximum number of uninterrupted rest periods during on-call duties.

The demand for physicians can be identified as the most common uncertain variable when looking at the physician scheduling literature in general (**?????**). **?** and **?** analyze the shift scheduling in an emergency department taking into account an uncertain patient arrival rate. While **?** use a SAA to solve their problem with 100 scenarios, **?** integrate a queuing system in their formulation using chance constraints. **?** develop a two-stage MIP assigning surgeries to anesthesiologists and operating rooms taking into account an uncertain duration of the surgeries. They use a robustness approach to define the surgery duration and measure the total overtime cost across all resources for a realization. **?** use a SAA to consider unexpected overtime in physicians daily schedules. They integrate their stochastic formulation in a column generation heuristic to decrease the unplanned overtime for a given workforce using 12 demand scenarios. While demand is subject to significant fluctuations when considering a short time horizon such as hours, this fluctuation has only an insignificant effect when considering a period length of days and weeks. The absence of individual employees, on the other hand, has an impact at this planning level and is considered in our work. A synthesis table summarizing the studied characteristics in literature with respect to an uncertain environment is given in Table 2.

The first column references the paper, while the time horizon (TH) of the problem is given in the second column. The next two columns account for the operational attributes duties (Dty) and demand (Dem). Duties refer to special shifts that affect the availability as defined in Section 1 and demand means that only a minimum or maximum number of workers can be assigned to the tasks under consideration, e. g., shifts or departments. The next three columns focus on the training attributes. The requirements (Req) usually define a minimum or maximum number of assignments to a specific task, e. g., shifts or departments. Priorities (Prio) refer to the concept introduced in Section 1 and continuity (Cont) refers to consecutive assignments to the same task. The next three columns highlight the different stochastic resources, i. e., demand (Dem), interventions (Int), and the availability (Avl). Note that the deterministic resident scheduling literature covers all attributes except for the priorities. The last two columns show information about the modeling approach as well as the type of distribution. The literature shows that SAA is a common approach to addressing such problems. Interestingly, the type of distribution changes in the different studies, even when the same aspect is considered, such as patient arrival rates.

Summarizing the literature review, we could identify several papers using training preferences as input for operational planning. These preferences can also be interpreted as priorities. However, there is no paper determining the preferences of residents nor assigning residents to rotations with different priorities as defined in Section 1, i. e., multiple assignments per period. While **?** and **?** do consider an uncertain environment for residents, there is no paper to the best of our knowledge, taking into account an uncertain availability of residents. In the literature that goes beyond resident planning problems, most papers consider demand uncertainty. From the literature review, we can conclude that we are closing several research gaps with this paper. We extend the stochastic formulations of the resident scheduling

Table 2: Literature synthesis indicating the studied characteristics in literature

		Operational attributes		Training attributes			Stochastics				
Paper	TH	Dty	Dem	Req	Prio	Cont	Dem	Int	Avl	Method	Distribution
?	6 weeks	x	x				x			Simulation	41
?	5 years		x	x				x		Γ-robustness	-
?	1 day		x				x			Chance constraint	Gamma
?	1 day		x				x			SAA	Poisson
?	1 day		x				x			Γ-robustness / SAA	- / Log-normal
?	1 week		x				x			SAA	Normal
?	1 month	x	x				x			SAA	Poisson
This paper	1 year	x	x	x	x	x			x	SAA	Uniform, Poisson

problem by the area of absences. Also, we are the first to model the possibility of creating (annual) training schedules with several assignments to departments using priorities.

3 Two-stage resident scheduling problem formulation

Uncertainty plays an important role in residents' annual training scheduling. As part of their training, they learn many different and demanding skills that they must perform independently within a very short time. In order to adequately prepare for new skills, residents need to know as early as possible what tasks they will face. Preparation is one of the reasons why there is an annual training schedule. Short-term changes in the daily schedule, such as absences, often lead to deviations from the training schedule. For example, in the real-world data of our case study for a whole year, about one-fifth of all daily assignments deviate from the training schedule. Unexpected department changes often mean that a resident has not enough time to prepare for the new assignment. A lack of preparation time can negatively influence the training and reduce employee satisfaction as well as the level of service, i. e., patient care. In order to increase the planning reliability of the training schedule, we present the following two-stage model formulation of the resident scheduling problem.

The planning horizon of this model is one year and will be described by the sets $w \in \mathscr{W}$ representing the weeks of a year and $t \in \mathscr{T}$ representing the days in a week. Note that the time horizon can be changed depending on the hospital's planning horizon, e. g., quarterly planning. Our main goal is to assign a set of residents $i \in \mathscr{I}$ to a set of departments $j \in \mathscr{J}$ on a weekly basis over the year so that residents have an overview of their assignments for the upcoming year, i. e., a tactical problem is solved. A set of priorities $p \in \mathscr{P}$ extends the dimension of assignments allowing more than one allocation to departments within a week in the training schedule. A duty and daily schedule is integrated to increase planning reliability, i. e., as an evaluation step to determine the department assignments with different priorities in the training schedule.

The first stage of our problem generates the annual training schedule. The core information of our model, included in the first stage, is given by the binary decision variable

$$x_{ijwp} = \begin{cases} 1, \text{ if resident } i \text{ is assigned to department } j \text{ in week } w \text{ with priority } p \text{ in the training schedule} \\ 0, \text{ otherwise} \end{cases}.$$

Here, an assignment to a department j with priority 1 corresponds to the resident's main department in week w (see definition in Section 1). Each resident has a lower $\underline{M}_{ij}^{\text{training}}$ and upper limit $\overline{M}_{ij}^{\text{training}}$ in terms of weeks of such assignments defined by the hospital management taking into account the training program and progress for the upcoming year. An upper bound may be useful if, for example, there are bottlenecks in certain interventions relevant to resident training in individual departments. For stable planning of the training schedule, the daily and duty schedule must also be taken into account (see Section 1). These are generated in the second stage of our problem. Therefore, in addition to the set of departments used for daily scheduling, a set of duties $d \in \mathscr{D}$ is needed. Some of the duties cannot be clearly assigned to a department in our use case, which is why we consider the duties independently of the department. The duties d as well as the departments j have a desirable range of residents per day t going from $\underline{D}_{dt}^{\text{duty}}$ to $\overline{D}_{dt}^{\text{duty}}$ and $\underline{D}_{jt}^{\text{daily}}$ up to $\overline{D}_{jt}^{\text{daily}}$ guaranteeing the functioning of the hospital. This range allows flexibility for the planner in terms of resident training goals. However, to be assigned to a duty or department, the resident must have the necessary seniority level given in the set $l \in \mathscr{L}$. Residents may switch the seniority level in the planning horizon, but the level is known in each period for each resident in advance for the upcoming year. Note that working on a weekend and overnight are handled in the duty schedule. For simplicity, we assume that a resident needs a day off after each duty. So, it is not possible to work several duties in a row. This means that each resident can work a maximum of 6 days a week. The assignment of residents to the duty and daily schedule is based on the absence of a resident $T^{\text{off}}_{iwt}(\omega)$ where ω is the random parameter for the second stage uncertainty. Remark that such uncertainties will be considered in an operational setting in a rescheduling problem and be re-solved daily. Nevertheless, these uncertainties affect the annual training schedule as well and should be considered in the creation process, i. e., a resident might not be assigned to the department of the training schedule on such a day. We are interested in generating a stable annual training schedule in the first place, i. e., residents should know in advance to which departments they might be assigned. A visual representation of the problem is given in Figure 1. The annual training schedule covers the entire planning horizon, i. e., it is published once. In contrast, a daily roster covers one month. The dotted arrows represent the direct influence of the annual training schedule (first stage decision) on the daily rosters (second stage decision). The recourse function $\mathscr{Q}(\mathbf{X}) := \mathbb{E}_\omega[Q(\mathbf{X}, \omega]$ evaluates the expected reward for assignments in the daily (R^{daily}) and duty schedule (R^{duty}) for a given training schedule $\mathbf{X}$ considering a reward R^{training}_p for each assignment in daily schedule according to the training schedule as well. Exactly this matching of the daily and training schedule is one of our main contributions. Note that our main target is to maximize these rewards. Please find a summary of all used notation in the following.

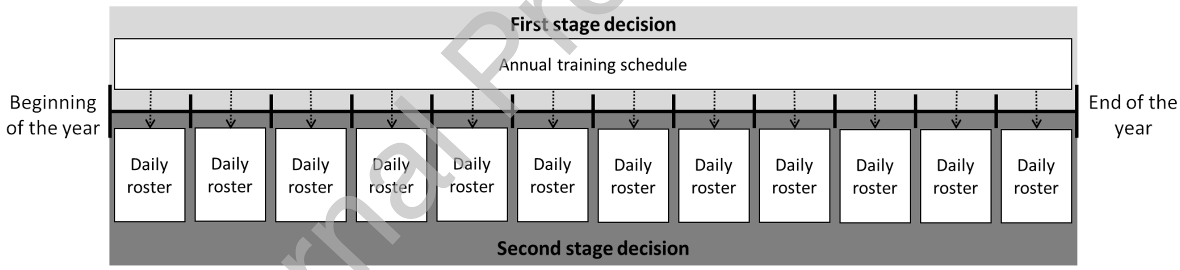

Figure 1: Visual representation of the two stage formulation

Sets with indices

$\mathscr{I}$	set of residents (index i)
$\mathscr{J}$	set of departments (index j)
$\mathscr{D}$	set of overnight duties (index d)
$\mathscr{W}$	set of weeks (index w)
$\mathscr{T}$	set of days in a week (index t)
$\mathscr{T}^{\text{work}}$	set of working days in a week ($\mathscr{T}^{work} \subseteq \mathscr{T}$)
$\mathscr{L}$	set of seniority level (index l)
$\mathscr{P}$	set of priorities (index p)

Parameters

ω	random parameter for the second stage uncertainty

$T^{\text{off}}_{iwt}(\omega)$	1, if resident i is absent on day t in week w, 0, otherwise
$L^{\text{resident}}_{ilw}$	1, if resident i has seniority level l in week w, 0, otherwise
L^{daily}_{jl}	1, if department j requires seniority level l, 0, otherwise
L^{duty}_{dl}	1, if overnight duty d requires seniority level l, 0, otherwise
G^{24h}	maximum number of overnight duties to be assigned to one resident in a single week
$\overline{M}^{\text{training}}_{ij}$	maximum number of weeks resident i should be assigned to department j in the training schedule
$\underline{M}^{\text{training}}_{ij}$	minimum number of weeks resident i should be assigned to department j in the training schedule
K_j	block length of department j in the training schedule
$\overline{D}^{\text{daily}}_{jt}$	maximum demand for residents in department j on day t in the daily schedule
$\underline{D}^{\text{daily}}_{jt}$	minimum demand for residents in department j on day t in the daily schedule
$\overline{D}^{\text{duty}}_{dt}$	maximum demand for residents on overnight duty d on day t in the duty schedule
$\underline{D}^{\text{duty}}_{dt}$	minimum demand for residents on overnight duty d on day t in the duty schedule
C^{daily}	cost per resident shortage on a department
R^{training}_{p}	reward for assigning a resident to a department as planned in the training schedule with priority p
R^{daily}	reward for assigning a resident to a department
R^{duty}	reward for assigning a resident to an overnight or weekend duty

Decision variables

x_{ijwp}	1, if resident i is assigned to department j in week w with priority p in the training schedule, 0, otherwise
$y_{ijwt}(\omega)$	1, if resident i is assigned to department j on day t in week w, 0, otherwise
$z_{idwt}(\omega)$	1, if resident i is assigned to duty d on day t in week w, 0, otherwise
$\delta^{\text{daily}}_{jwt}(\omega)$	resident deficit for department j on day t in week w

$$Q(\mathbf{X},\omega) = \max \sum_{i\in\mathscr{I}}\sum_{j\in\mathscr{J}}\sum_{w\in\mathscr{W}}\sum_{t\in\mathscr{T}}\sum_{p\in\mathscr{P}} R^{\text{training}}_{p} x_{ijwp} y_{ijwt}(\omega) \tag{1a}$$

$$+\sum_{i\in\mathscr{I}}\sum_{j\in\mathscr{J}}\sum_{w\in\mathscr{W}}\sum_{t\in\mathscr{T}} R^{\text{daily}} y_{ijwt}(\omega) + \sum_{i\in\mathscr{I}}\sum_{d\in\mathscr{D}}\sum_{w\in\mathscr{W}}\sum_{t\in\mathscr{T}} R^{\text{duty}} z_{idwt}(\omega) \tag{1b}$$

$$-\sum_{j\in\mathscr{J}}\sum_{w\in\mathscr{W}}\sum_{t\in\mathscr{T}} C^{\text{daily}} \delta^{\text{daily}}_{jwt}(\omega) \tag{1c}$$

Objective function. The objective function (1) can be divided into three parts. While all three are dependent on the second stage decision, the first one also depends on the first stage. In (1a) we want to maximize the reward of having residents assigned to their department from the training schedule (x_{ijwp}) in the daily schedule ($y_{ijw}(\omega)$). This term is used to measure the implementation of the training schedule at the operational level. Note that the decision variables x_{ijwp} and $y_{ijwt}(\omega)$ are multiplied together so that the objective function is non-linear. Since non-linear functions often make it difficult to find a solution, we will linearize the term after the complete model is described. By using different rewards for the different priorities, we want to facilitate the assignment to the more important priority, i. e., $R_p^{\text{training}} > R_{p'}^{\text{training}}$ if $p < p'$. Typically, hospital management sets rewards for each priority to ensure training progress. Note, these can also be individualized for residents to give preferences on the respective training areas, i. e., by extending the parameter to $R_{ijwp}^{\text{training}}$. However, for the subject under investigation, this extension does not provide any value. The second part (1b) of the objective function rewards the assignment of residents to departments in the daily schedule ($y_{ijwt}(\omega)$) as well as to duties in the duty schedule ($z_{idwt}(\omega)$). Note, R^{daily} and R^{duty} are not individualized, as we do not generate actual daily and duty schedules in the model but use them to build the stable annual training schedule. As said in Section 1, daily and duty schedules are published monthly (see **??**). In the last term (1c) we penalize violations on an operational level. In particular, we look at the shortfall in the minimum demand per department in the daily schedule. Note that the rewards and costs are primarily weights to control the construction of the different schedules. However, in a real-world setting, it is usually the case that $R^{\text{duty}} \gg C^{\text{daily}} \gg R^{\text{daily}}, R_p^{\text{training}}$ (**?**). The reasons are primarily to maintain operations 24/7 and, along with that, to ensure health care and the hospital's profitability.

$$\sum_{j\in\mathscr{J}} x_{ijwp} \leq 1 \qquad \forall i \in \mathscr{I}, w \in \mathscr{W}, p \in \mathscr{P} \tag{2}$$

$$\sum_{w\in\mathscr{W}} \sum_{p\in\mathscr{P}} x_{ijwp} \leq \overline{M}_{ij}^{\text{training}} \qquad \forall i \in \mathscr{I}, j \in \mathscr{J} \tag{3}$$

$$\sum_{w\in\mathscr{W}} x_{ijw1} \geq \underline{M}_{ij}^{\text{training}} \qquad \forall i \in \mathscr{I}, j \in \mathscr{J} \tag{4}$$

$$\sum_{j\in\mathscr{J}} x_{ijwp} - \sum_{j\in\mathscr{J}} x_{ijwp'} \geq 0 \qquad \forall i \in \mathscr{I}, w \in \mathscr{W}, p, p' \in \mathscr{P}, p < p' \tag{5}$$

$$x_{ijwp} + x_{ijwp'} \leq 1 \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, w \in \mathscr{W}, p, p' \in \mathscr{P}, p < p' \tag{6}$$

$$x_{ijw'p} \geq x_{ijwp} - x_{ijw-1p} \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, w \in \mathscr{W} \setminus \{|\mathscr{W}| - K_j + 1, \ldots, |\mathscr{W}|\}, p \in \mathscr{P}, w' \in \{w+1, \ldots, w+K_j-1\} \tag{7}$$

$$x_{ijw'p} \geq x_{ij1p} \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, p \in \mathscr{P}, w' \in \{2, \ldots, K_j\} \tag{8}$$

Training schedule. The Constraints (2) up to (6) are constructing the training schedule and are the first stage decision. Constraints (2) ensure that a resident can be assigned to at most one department per week and priority. An individual upper bound of assignments to a specific department in the training schedule is given in Constraints (3). Note that this bound considers all priorities, i. e., a resident can be assigned to one department at most $\overline{M}_{ij}^{\text{training}}$ times, regardless of the priority. However, the lower bound of assignments to one department in the training schedule is limited to priority 1 as stated in Constraints (4). Under the assumption that a resident is primarily assigned to their first priority, the minimum training progress should be ensured in this respect. Note that $\overline{M}_{ij}^{\text{training}}$ and $\underline{M}_{ij}^{\text{training}}$ are defined individually per resident. The main reason for this is that the training goals may differ between the individual residents for the upcoming year. This difference is mainly but not exclusively due to the different training years of the residents, i. e., even within the same training year, residents can have a different focus. The priorities have an order to each other, and this is ensured by Constraints (5), i. e., a resident can only be assigned to a department with priority 2 if they are also assigned to a department with priority 1 in the same week. Additionally, we have to ensure that a resident is not assigned to the same department with different priorities in the same week as described in Constraints (6). Finally, the training schedule should be used at the operational level to enforce continuity, i. e., to work consecutive weeks in the same department. Therefore, Constraints (7) and (8) ensure that an assignment in the training schedule to a department j is at least K_j weeks. The idea of these constraints is to design blocks of training. For example, let parameter $K_j = 4$. In this case, a resident who is assigned in the first week of the year to department A must be assigned to department A in the following three weeks as well. These blocks are usually used to increase the learning effect of a resident in one department as well as to stabilize the service level of the departments (**?**).

$$\sum_{i \in \mathscr{I}} y_{ijwt}(\omega) \leq \overline{D}_{jt}^{\text{daily}} \qquad \forall j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T} \tag{9}$$

$$\sum_{i \in \mathscr{I}} y_{ijwt}(\omega) + \delta_{jwt}^{\text{daily}}(\omega) \geq \underline{D}_{jt}^{\text{daily}} \qquad \forall j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T} \tag{10}$$

$$\sum_{j \in \mathscr{J}} y_{ijwt}(\omega) \leq 1 - T_{iwt}^{\text{off}}(\omega) \qquad \forall i \in \mathscr{I}, w \in \mathscr{W}, t \in \mathscr{T} \tag{11}$$

$$y_{ijwt}(\omega) \leq \sum_{l \in \mathscr{L}} (L_{jl}^{\text{daily}} L_{ilw}^{\text{resident}}) \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T} \tag{12}$$

Daily schedule. The block of Constraints (9) to (12) is used to construct the daily schedule for the complete time horizon and is the first part of the second-stage decision. Note that the assignment for the daily schedule depends on the realization of the uncertain term. At the operational level a minimum and maximum number of residents can be assigned to a department as formulated in Constraints (9) and (10). It is also possible to assign fewer residents to a department than the minimum required. However, such a violation is penalized in the objective function (see (1c)). Constraints (11) ensure that a resident can be assigned to a maximum of one department per day, if available on that day ($T_{iwt}^{\text{off}}(\omega)$). Note that the availability is the uncertain parameter. Additionally, Constraints (12) ensure that a resident's assignment to a department is possible if the resident has the seniority level needed by the department. Note that this type of formulation also allows a resident with a higher seniority level to be assigned to a department with a lower seniority level requirement. For instance, a seniority level l can remain active ($L_{ilw}^{\text{resident}} = 1 \ \forall w \in W : w \geq w'$) after a resident has reached the next year of training in the period w'.

$$\sum_{i \in \mathscr{I}} z_{idwt}(\omega) \leq \overline{D}_{dt}^{\text{duty}} \qquad \forall d \in \mathscr{D}, w \in \mathscr{W}, t \in \mathscr{T} \tag{13}$$

$$\sum_{i \in \mathscr{I}} z_{idwt}(\omega) \geq \underline{D}_{dt}^{\text{duty}} \qquad \forall d \in \mathscr{D}, w \in \mathscr{W}, t \in \mathscr{T} \tag{14}$$

$$\sum_{d \in \mathscr{D}} z_{idwt}(\omega) \leq 1 - T_{iwt}^{\text{off}}(\omega) \qquad \forall i \in \mathscr{I}, w \in \mathscr{W}, t \in \mathscr{T} \tag{15}$$

$$z_{idwt}(\omega) \leq \sum_{l \in \mathscr{L}} (L_{dl}^{\text{duty}} L_{ilw}^{\text{resident}}) \qquad \forall i \in \mathscr{I}, d \in \mathscr{D}, w \in \mathscr{W}, t \in \mathscr{T} \tag{16}$$

$$\sum_{d \in \mathscr{D}} \sum_{t \in \mathscr{T}} z_{idwt}(\omega) \leq G^{24\text{h}} \qquad \forall i \in \mathscr{I}, w \in \mathscr{W} \tag{17}$$

$$\sum_{d \in \mathscr{D}} 3 z_{idw(t-1)}(\omega) \leq 3 - \sum_{d \in \mathscr{D}} z_{idwt}(\omega) - \sum_{j \in \mathscr{J}} y_{ijwt}(\omega) - T_{iwt}^{\text{off}}(\omega) \qquad \forall i \in \mathscr{I}, w \in \mathscr{W}, t \in \mathscr{T}, t > 1 \tag{18}$$

$$\sum_{d\in\mathscr{D}} 3z_{id(w-1)7}(\omega) \leq 3 - \sum_{d\in\mathscr{D}} z_{idw1}(\omega) - \sum_{j\in\mathscr{J}} y_{ijw1}(\omega) - T^{\text{off}}_{iw1}(\omega) \qquad \forall i\in\mathscr{I}, w\in\mathscr{W}, w>1 \quad (19)$$

Duty schedule. The duty schedule extends the daily schedule by assigning overnight duties ($\mathscr{D}$) and ensures treatment 24/7. Note that the assignment of duties is also part of the second-stage decision. Constraints (13) and (14) limit the number of residents that can be assigned to a duty per day. Constraints (15) and (16) are analogous to Constraints (11) and (12) of the daily schedule and ensure that a resident can be assigned to no more than one duty per day if available and that they have the appropriate seniority level. Overnight duties are stressfull to residents as they require the resident to be present in the hospital for 24 hours. Therefore the total number of overnight duties in one week is limited for every resident by Constraints (17). The next two constraints ensure that a resident has a day off after being assigned to an overnight duty, i. e., no assignment to a duty ($z_{idwt}(\omega) = 0$) nor a regular shift ($y_{ijwt}(\omega) = 0$). Additionally, a resident should not be assigned to a duty if they have already a day off ($T^{\text{off}}_{iwt}(\omega)$) on the following day. While Constraints (18) consider the days from Monday up to Saturday, Constraints (19) are needed for overnight duties on Sunday ($t = 7$). Eventually, variable domain definitions are handled in Constraints (20) and (21).

$$x_{ijwp}, y_{ijwt}(\omega), z_{idwt}(\omega) \in \{0,1\} \qquad \forall i\in\mathscr{I}, j\in\mathscr{J}, d\in\mathscr{D}, w\in\mathscr{W}, t\in\mathscr{T}, p\in\mathscr{P} \qquad (20)$$

$$\delta^{\text{daily}}_{jwt}(\omega) \in \mathbb{Z}_+ \qquad \forall i\in\mathscr{I}, j\in\mathscr{J}, d\in\mathscr{D}, w\in\mathscr{W} \qquad (21)$$

The solution of this formulation will result in a feasible annual training schedule taking into account the operational level of a daily schedule as well as the duty schedule, i. e., a weekly assignment of residents to departments with different priorities. Remark that the solution of the daily and duty schedule cannot be used directly for the operational level, i. e., a duty schedule is usually published on a monthly basis in order to consider individual preferences such as working a night duty on Saturday. In addition, actual absences are not known in advance. However, the hospital management can use the solution of the duty schedule as well as the daily schedule provided by our model as a start solution for operational planning. In general, solving non-linear programs is much harder than solving linear ones. Therefore, we will linearize the non-linear term of the objective function (1a) in the following paragraph.

Linearization of the non-linear objective term. The non-linear term $x_{ijwp} \cdot y_{ijwt}(\omega)$ in the objective function (1a) can be expressed in a linear form by using additional binary decision variables $\pi_{ijwtp}(\omega) \in \{0,1\}$. Note, $\pi_{ijwtp}(\omega)$ is defined as a binary variable since x_{ijwp} and $y_{ijwt}(\omega)$ are binary as well. Consequently, the multiplication of the two variables can also be only binary. As a first result, we can replace the objective term (1a) by the following term (22a).

$$\max \sum_{i\in\mathscr{I}} \sum_{j\in\mathscr{J}} \sum_{w\in\mathscr{W}} \sum_{t\in\mathscr{T}} \sum_{p\in\mathscr{P}} R^{\text{training}}_{p} \pi_{ijwtp}(\omega) \qquad (22a)$$

Additionally, we need to ensure that $\pi_{ijwtp}(\omega)$ becomes equal to 1 if and only if $x_{ijwp} = y_{ijwt}(\omega) = 1$. For this purpose, three additional types of constraints are required.

$$\pi_{ijwtp}(\omega) \leq x_{ijwp} \qquad \forall i\in\mathscr{I}, j\in\mathscr{J}, w\in\mathscr{W}, t\in\mathscr{T}, p\in\mathscr{P} \qquad (23)$$

$$\pi_{ijwtp}(\omega) \leq y_{ijwt}(\omega) \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T}, p \in \mathscr{P} \qquad (24)$$

$$\pi_{ijwtp}(\omega) \geq x_{ijwp} + y_{ijwt}(\omega) - 1 \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T}, p \in \mathscr{P} \qquad (25)$$

$$\pi_{ijwtp}(\omega) \in \{0,1\} \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T}, p \in \mathscr{P} \qquad (26)$$

Constraints (23) and (24) bound $\pi_{ijwtp}(\omega)$ by the corresponding training schedule variable x_{ijwp} (first stage) and daily schedule variable $y_{ijwt}(\omega)$ (second stage), i. e., $\pi_{ijwtp}(\omega) = 0$ if $x_{ijwp} = 0$ or $y_{ijwt}(\omega) = 0$. Constraints (25) are needed to ensure that $\pi_{ijwtp}(\omega) = 1$ if both x_{ijwp} and $y_{ijwt}(\omega)$ are equal to 1. Finally, variable domain definition is handled in Constraints (26).

4 Sample average approximation for resident scheduling

In this section, we first transfer the uncertainty of the second stage of our problem into a scenario approach and then decompose the problem to solve it efficiently. With a planning horizon of one year, uncertainty can be replicated by an absence plan, i.e., each scenario describes the absence of all residents for the entire planning horizon. For a detailed description of the SAA, we recommend **?**. We assume that the random parameter ω follows a discrete distribution with a finite support. Therefore, let $s \in \mathscr{S}$ be the set of scenarios and $p_s > 0$ the probability for a realization of scenario ω_s. Note that $\sum_{s\in\mathscr{S}} p_s = 1$. In this case, $\mathscr{Q}(\mathbf{X}) = \mathbb{E}_\omega[Q(\mathbf{X},\omega)] = \sum_{s\in\mathscr{S}} p_s Q(\mathbf{X},\omega_s)$ applies to our recourse function. Consequently, the second stage decision variables $y_{ijwt}(\omega), z_{idwt}(\omega), \delta^{\text{daily}}_{jwt}(\omega), \pi_{ijwtp}(\omega)$ and the random absences $T^{\text{off}}_{iwt}(\omega)$ will be extended by the set of scenarios. The complete model formulated as a sample average approximation resident scheduling problem (SAA-RSP) can be found in the appendix. The realization of the absences of all residents over the year defines a scenario, i. e., $T^{\text{off}}_{iwts} = 0 \vee 1$ (present or absent). The number of scenarios can quickly become very large, so it is not realistic to map all scenarios. Instead, we will use a subset of scenarios to obtain a good estimator (**?**). In general, looking at the size of the formulation, we find that – even in the deterministic case, i. e., taking only one scenario into account – the problem is too big to be solved by standard solvers. The real-world instance encountered in this paper with $|\mathscr{I}| = 80$ residents, $|\mathscr{J}| = 14$ departments, and $|\mathscr{D}| = 17$ overnight duties leads to more than $1.8 \cdot 10^6$ binary decision variables and $3.6 \cdot 10^6$ constraints assuming only two priorities in the training schedule. Note that each additional scenario taking into account increases the problem size by almost the same amount.

A common approach in the literature to deal with such large problem sizes and scenarios is decomposing the problem (**?**). Analyzing the mathematical formulation of Section 3, we can see that the first stage decision, i. e., the constraints for the training schedule, is connected with the second stage only by the objective (1a). Note, Constraints (25) are the connection after linearization of the non-linear objective term. We can take advantage of this characteristic and decompose the problem precisely between the first and second stages (see Figure 2). Note that the domain definitions of the decision variables stay the same as well. Additionally, we know that the scenarios $s \in \mathscr{S}$ of the second stage are independent of each other. As a result, we can solve each scenario individually. To solve the decomposition, we propose an iterative approach where the first and second stages are solved alternately as shown in Algorithm 1. For solving the two stages independently, the linking variables π must be manipulated based on the solution of the first or second stage of the problem. We know from the formulation in Section 3 that the objective function term (22a) can only be changed when solving the second stage if an assignment has taken place

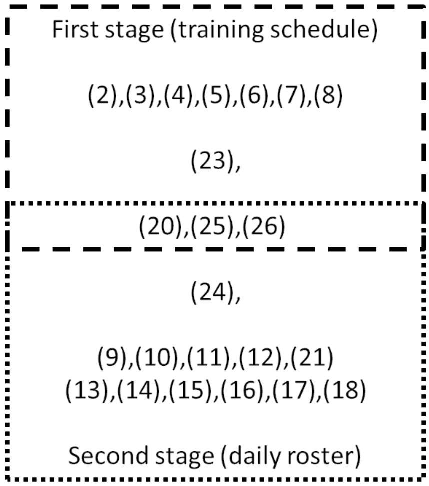

Figure 2: Visualization of the constraints for the decomposed two stage resident scheduling formulation

in the first stage, i. e., the upper bound for π is determined by the $\mathbf{x}$ variables of the first stage. The same applies to the first stage – only if someone has been assigned to a department in the daily schedule of the second stage, the assignment of the first stage influences the objective function term (22a), i. e., the upper bound for π when solving the first stage depends on the individual $\mathbf{y}$ variables of the scenarios. In what follows, we refer to this relationship with the upper bound of π depending on which stage we are considering. The π variables are initialized with 0. The termination criteria can be set by the user, e. g., by time or optimality gap. However, the algorithm finally terminates if the first stage's objective value and those of the second stage do not change anymore within one iteration. Note that our decomposition can also be seen in the sense of an L-shaped approach, where we do not explicitly formulate the cuts from the second stage and only use them for a single iteration (**?**).

For the first stage, the decomposition and fixation of the bounds have the consequence that the A-matrix is totally unimodular and can be solved efficiently as a linear program. Remark that we can formulate the first stage as a circulation problem with integral capacities, lower bounds on the flow, and node requirements. In this formulation each decision variable x_{ijwp} can be seen as an edge of the graph. We can further reduce the formulation to a maximum flow problem (**?**). The upper bound of the π variables determine the weight of the edges x_{ijwp}. Since the upper bound is determined from the daily schedule in the second stage of scenario $s \in \mathscr{S}$ it can be replaced by $\sum_{s\in\mathscr{S}}\sum_{t\in\mathscr{T}} p_s R_p^{\text{training}} y_{ijwts}$. The objective value of this formulation corresponds to the objective (22a). Objective (1b) and (1c) are a constant in the first stage and can be added after the optimization.

For the second stage, we propose two solution strategies that we will evaluate in the experimental study of Section 5. In the first case, we solve all sample scenarios after terminating the new bounds for π, i. e., the generated subset of the complete sample. This gives us an operational assignment (daily and duty schedule) for each scenario and a complete evaluation for the training schedule, i. e., $\mathscr{Q}(\mathbf{X})$. Although the individual scenarios can be solved independently and in parallel, the computing effort in an actual application inevitably increases as the number of scenarios increases, i. e., the solution time over all scenarios. Therefore we consider a second approach where we do not solve all scenarios in every iteration but only a subset $\mathscr{B} \subset \mathscr{S}$. We will refer to this as the batching method. With this variation, the solution time within one iteration can be shortened, i. e., if not all scenarios can be solved in parallel.

Algorithm 1 Solution algorithm for the SAA-RSP

```
\* Initialization *\
Initialize the first stage problem (1) and the second stage problem (2) for all scenarios s ∈ 𝒮. Set 𝒮̄ = 𝒮.
while Termination criteria is False do
    \* Generate the training schedule *\
    Solve problem (1).
    \* Update π based on the training schedule *\
    Set π.ub = x.
    \* Choose a subset of scenarios *\
    if batching is True then
        Randomly choose a subset 𝒮̄ = ℬ ⊂ 𝒮
    end if
    for s ∈ 𝒮̄ do
        \* Generate the daily roster for scenario s *\
        Solve problem (2).
    end for
    \* Update π based on the daily rosters *\
    Set π.ub = y.
end while
Report the final solution.
```

One of the advantages of this iterative solution approach is that in each iteration, a feasible solution is generated, i. e., a lower bound. Consequently, the hospital management can choose between several schedules. A major drawback, however, is that we do not get an upper bound within the algorithm. Nevertheless, the algorithm moves towards a local optimum over the iterations, since ensuring operational functionality is given higher weight than adherence to the training schedule.

To overcome the problem with the upper bound in our algorithm, we have analyzed ways to obtain a strong bound efficiently. One problem for determining a strong dual bound is the size of the compact model. Additionally, the LP relaxation offers only a poor bound with respect to the optimal objective value, and in the deterministic case alone takes more than 300 seconds to solve for realistic problem sizes. One reason for the poor performance of the bound is due to the Constraints (18) and (19). In the LP solution it is possible to be assigned to the daily schedule after a duty if more than one resident is assigned to the duty ($z_{idwts} \leq 0.5$). However, this is not the only reason. In the experimental study, we will see that the bound increases with additional priorities as well. In our preliminary studies of this problem, we found out that a Lagrangian relaxation of Constraints (4) could give a strong dual bound. Nevertheless, a solution time of more than eight hours for a small number of scenarios is beyond the scope. Therefore, we first wanted to determine an analytical bound (UB^{AB}) for the deterministic case as shown in Proposition 1.

Proposition 1. *An upper bound for the deterministic resident scheduling problem ($\mathbf{T}^{off} \triangleq$ known absences or $\mathbf{0}$) can be determined by the reward for all duties in the time horizon, the reward for all residents being assigned to a department in the time horizon, and the reward for all residents being assigned to their first priority, i. e., $z^* \leq R^{duty}\alpha + R^{daily}\beta + R_1^{training}\gamma$.*

Proof. Let α be the total number of duties in the time horizon, β the total number of residents being assigned to a department in the time horizon, and γ the total number of residents being assigned to their first priority of the training schedule.

(1.) Due to the practical characteristics of the resident scheduling problem, each duty is allocated first, i. e., $R^{\text{duty}} \gg C^{\text{daily}} \gg R^{\text{daily}}, R_p^{\text{training}}$. As a result, α is bounded by $|\mathscr{W}| \sum_{d,t} \overline{D}_{dt}^{\text{duty}}$.

(2.) On each working day at most $|\mathscr{I}|$ residents can be assigned to a department. However, Constraints (18) and (19) ensure that a resident has a day off if he or she was on duty the day before. Since $R^{\text{duty}} \gg C^{\text{daily}} \gg R^{\text{daily}}$ the number of residents can be reduced by the number of duties for which the following day (day off) is a working day. Additionally, we can reduce the value by the total number of known absences. Thus, β is bounded by $|\mathscr{I}||\mathscr{T}^{\text{work}}||\mathscr{W}| - |\mathscr{W}| \sum_{d,t} \overline{D}_{dt}^{\text{duty}} - |\mathbf{T}^{\text{off}}|$ with $t \in \{k \in \mathscr{T} | (k+1) mod(7) \in \mathscr{T}^{\text{work}}\}$.

(3.) We can assume that all assignments of (2.) are the first priority in the training schedule of the residents, i. e., $\gamma = \beta$.

Then follows $z^* \leq R^{\text{duty}}\alpha + R^{\text{daily}}\beta + R_1^{\text{training}}\gamma$ with the bounds for α, β, γ given in (1.),(2.), and (3.).

□

Since the bound can also be calculated for individual scenarios (see (2.) in Proposition 1), it is also possible to generate an upper bound for instances with several scenarios, i. e., $UB^{AB} = \sum_{s \in S} p_s UB_s^{AB}$ with UB_s^{AB} the analytical bound of scenario s. Additionally, this bound is valid for any number of priorities since we assume all assignments to be the first priority in the training schedule. In the experimental study, we show that the analytical bound is, on average, more than 8% better than the LP relaxation. Nevertheless, the study from Section 5 shows that even this bound is still a poor bound, i. e. the upper bound may be tighter. For this reason, we approximate an additional upper bound based on the solution methodology. Since the deterministic case can be solved quickly with our approach, we wanted to define a bound based on the deterministic solution under the assumption that $\mathbf{T}^{\text{off}} \triangleq$ known absences or $\mathbf{0}$. Note that the deterministic solution is a lower bound. However, we can assume that the algorithm cannot find a better solution in any scenario. As explained in Proposition 2, the upper bound ($UB^{SAA-RSP}$) can be defined by $z^* \leq \zeta \sum_{s \in \mathscr{S}} p_s(\hat{z} - R^{\text{daily}} |\mathbf{T}^{off}(\omega_s)| - \sum_{p \in \mathscr{P}} R_p^{\text{training}} \varepsilon_p(\omega_s))$ where ζ is the approximation level. Note that we will estimate ζ in the experimental study in Section 5

Proposition 2. *An upper bound for the SAA-RSP can be determined by solving the deterministic model ($\mathbf{T}^{off} \triangleq$ known absences or $\mathbf{0}$) and the random absences $T_{iwt}^{off}(\omega_s)$ in the scenarios $s \in \mathscr{S}$ with $z^* \leq \zeta \sum_{s \in \mathscr{S}} p_s(\hat{z} - R^{daily} |\mathbf{T}^{off}(\omega_s)| - \sum_{p \in \mathscr{P}} R_p^{training} \varepsilon_p(\omega_s))$.*

Proof. Assume $\hat{z}$ be the optimal objective value for the deterministic model, and $|\mathbf{T}^{off}(\omega_s)|$ the amount of absences in a scenario.

(1.) The SAA-RSP is bounded by $\hat{z}$. Obviously, $z^* \leq \sum_{s \in \mathscr{S}} p_s \hat{z}$ since the solution space is decreased, i. e., Constraints (11), (15), (18), and (19) decrease the solution space if $T_{iwt}^{off}(\omega_s) = 1$ for any $i \in \mathscr{I}, w \in \mathscr{W}, t \in \mathscr{T}$.

(2.) The number of available residents per day in each scenario is less or equal to the number of available residents in the deterministic model. Consequently, the rewards for assigning residents in the daily schedule decreases by the total amount of absent residents, i. e., $z^* \leq \sum_{s\in\mathscr{S}} p_s(\hat{z} - R^{\text{daily}}|\mathbf{T}^{off}(\omega_s)|)$.

(3.) The absence of the residents also affects the reward for assignments to the training schedule. We know from the deterministic solution how often a resident is assigned to a department and with which priority (and with no priority) based on the training schedule. $\varepsilon_p(\omega_s)$ is the total number of residents that will no longer be assigned to a department with priority p in scenario s. $\varepsilon_p(\omega_s)$ can be calculated by Algorithm 2. Then follows $z^* \leq \sum_{s\in\mathscr{S}} p_s(\hat{z} - R^{\text{daily}}|\mathbf{T}^{off}(\omega_s)| - \sum_{p\in\mathscr{P}} R_p^{\text{training}} \varepsilon_p(\omega_s))$.

If we now assume that $\hat{z}$ is not optimal, i. e., it is only a lower bound (obtained by our solution algorithm), then a parameter ζ can be determined, which can estimate an upper bound for all instances, i.e, $z^* \leq \zeta \sum_{s\in\mathscr{S}} p_s\hat{z} \rightarrow \zeta \geq \frac{z^*}{\sum_{s\in\mathscr{S}} p_s\hat{z}}$. Note that this is an ex-post analysis. The bound can only be calculated after solving the deterministic model.

□

Algorithm 2 Calculation of the priority reduction per scenario $\varepsilon_p(\omega_s)$

Let $|\mathbf{T}^{off}(\omega_s)|$ the amount of absences in a scenario and n_p the total number of residents assigned to a department as stated in the training schedule with priority p ($\pi_{ijwtp} = 1$) where n_0 is the total number of residents assigned to a department without any priority of the deterministic model.
Set $\varepsilon_0(\omega_s) = \min\{n_0, |\mathbf{T}^{off}(\omega_s)|\}$, $\varepsilon_p = 0\ \forall p \in \mathscr{P}$, $\alpha = \max\{0, |\mathbf{T}^{off}(\omega_s)| - n_0\}$, and $\beta = |\mathscr{P}|$.
while $\alpha > 0$ and $\beta \neq 0$ **do**
  Set $\varepsilon_\beta(\omega_s) = min\{n_\beta, \alpha\}$.
  Set $\alpha = \max\{0, \alpha - n_\beta\}$.
  Set $\beta = \beta - 1$.
end while
Report $\varepsilon(\omega_s)$.

Determining an upper bound in this way has several advantages. First, we can derive an upper bound for each scenario. We can integrate these bounds into each model of the second stage to accelerate the solution process. Second, we can derive a global upper bound for the overall problem. Herewith, we can evaluate the solution quality of the algorithm in each iteration.

5 Analyzing training priorities in annual scheduling

In this section, we apply our model in an experimental study based on real-world data concerning residents in anesthesiology of a large teaching hospital in Germany with more than 1,200 beds. In Section 5.1, we will analyze the performance of our developed solution approach. After that we will derive managerial insights based on parameter variations of our problem in Section 5.2.

In our study, we analyze the annual scheduling of anesthesiology residents, i. e., $|\mathscr{W}| = 52$ and $|\mathscr{T}| = 7$ resulting in a total of 364 planning days. Residents can be assigned in the training schedule and daily schedule to 14 different departments. Additionally, 17 different overnight duties with a total of 86 assignments per week must be considered in the duty schedule, i. e., there is not a demand for every duty every day. The absences for the scenarios are drawn equally distributed with a probability of 10% per day and resident. We defined the absentee rate in consultation with the hospital and in consideration of the literature (?). We have also tested the algorithm with other distributions and obtained similar results, e. g., a Poisson distribution for sequences of absences. An overview of the relevant parameters is given in Table 3 and the complete information is given in the online supplementary.

Table 3: Model information for the experimental study

$\lvert\mathscr{I}\rvert$	$\lvert\mathscr{J}\rvert$	$\lvert\mathscr{D}\rvert$	$\lvert\mathscr{W}\rvert$	$\lvert\mathscr{T}\rvert$	$\lvert\mathscr{L}\rvert$	G^{24h}	R^{daily}	R^{duty}	C^{daily}
80	14	17	52	7	5	2	10	200	50

All computations are performed on a 1.9 GHz (Intel® Core™ CPU i7-8650U) with 16 GB RAM running under the Windows 10 Enterprise operating system. All models are coded in Python 3.7, and Gurobi 9.0 (?) is used to solve all instances of the formulations. The default settings of Gurobi are used unless otherwise stated in the specific section.

5.1 Evaluation of the solution approach

In this Section, we will analyze the performance of the solution approach. For this purpose, we will first evaluate the performance of our upper bound as described in Section 4 and compare it with the LP relaxation of the compact formulation as well as a combinatorial relaxation of Constraints (4) since the Lagrangian relaxation is out of scope (see Section 4), i. e., the Lagrangian multiplier is fixed to 0 (?). We will then evaluate the performance of our SAA-RSP algorithm in terms of the number of scenarios. Here we will analyze both options of the algorithm, i. e., with all scenarios in each iteration and with batches.

Performance of the upper bound. The upper bound is an important tool for measuring solution quality. In this part of the study, we evaluate the performance of the LP relaxation of the model presented in Section 3, a combinatorial relaxation of Constraints (4), the analytic bound (see Proposition 1), and the bound derived from the solution algorithm (see Proposition 2). Finally, we estimate the parameter ζ based on the evaluation (see Proposition 2).

For the determination of an upper bound, we have set a time limit of one hour. This time limit reduces this part of the analysis to at most 1 scenario and 2 priorities. For larger sets, the LP relaxation already exceeds the time limit. We analyzed in total 5 different instances with a varying number of known absences based on the real-world data using one and two priorities each with 10 different scenarios, i. e., 110 settings (10 deterministic and 100 with a scenario). The results of the deterministic instances with $|\mathscr{P}| = 1$ and $|\mathscr{P}| = 2$ are given in Table 4.

The first two columns identify the instance and the number of priorities. The next two columns show the optimal solution of the LP relaxation (upper bound of the original problem) and the solution time. Here it is noticeable that the objective function value increases with two possible priorities compared to one. The same information is given in columns five and six but for the combinatorial relaxation. While the combinatorial relaxation (CR) is the strongest of the three upper bounds presented, it should be noted that it can only be solved optimally with one priority. In other instances, the time limit was exceeded, and the current upper bound at termination was used accordingly. The analytic bound (AB) is given in the seventh column. Here we can see directly that the bound between one and two priorities does not change. The reason for this is that when calculating the bound, it is assumed that all assignments

Table 4: Deterministic results for the resident scheduling problem

Instance	$\lvert\mathscr{P}\rvert$	LP relaxation UB	LP relaxation Time (s)	Combinatorial relaxation UB	Combinatorial relaxation Time (s)	Analytic bound UB	SAA-RSP algorithm LB	SAA-RSP algorithm Iterations	SAA-RSP algorithm Time (s)
1	1	1,527,232	215	1,332,580	511	1,421,280	1,282,930	19	404
2	1	1,525,710	267	1,330,560	310	1,420,160	1,278,910	13	259
3	1	1,527,032	299	1,331,760	359	1,420,560	1,285,480	19	480
4	1	1,534,339	217	1,345,570	333	1,426,800	1,297,620	22	521
5	1	1,528,012	183	1,337,720	292	1,423,400	1,287,580	20	557
1	2	1,601,147	623	1,398,289	3,601	1,421,280	1,305,835	25	512
2	2	1,599,516	586	1,396,065	3,601	1,420,160	1,304,895	26	535
3	2	1,600,962	354	1,397,340	3,601	1,420,560	1,305,305	18	422
4	2	1,609,049	494	1,412,155	3,601	1,426,800	1,319,355	24	508
5	2	1,602,174	381	1,403,770	3,601	1,423,400	1,310,050	20	558

are carried out with the first priority. Consequently, as the number of priorities increases, the analytical bound becomes increasingly stronger than the LP bound, i. e., about 7% better with one priority and almost 12% better using two priorities. The last four columns show the information of the SAA-RSP algorithm. The optimality gap between the best upper bound (combinatorial relaxation) and the solution of the SAA-RSP algorithm is on average 5%. Note that for the SAA-RSP algorithm no upper bound is given since we want to estimate the bound in a next step for the scenarios.

We first analyze the solution time of each approach for the 100 instances. We differentiate between the settings using one and two priorities. For the SAA-RSP algorithm, we used the deterministic cases' solution time since this value is the basis for the calculation. The scenario-specific values as described in Proposition 2 can be added at almost no cost, i. e., it takes less than 1 second. The LP relaxation's average solution time using one and two priorities is significantly different (p-value= 0.0018), i. e., the solution time is increasing with additional priorities. With a p-value of 0.0002, the SAA-RSP algorithm's average solution time is significantly different as well for one and two priorities. However, we have to consider that these 100 instances' solution time is based on only ten instances. Consequently, the sample is too small to evaluate the difference in solution time for the SAA-RSP algorithm. We will take a closer look at the solution time when we analyze the performance of the algorithm. For the CR, it can be stated that instances with two priorities can no longer be solved within an hour.

Solving the CR not optimal has consequences on the quality of the bounds. To evaluate the quality of the bounds, we have taken the LP relaxation as a basis, i. e., $\frac{UB^b}{UB^{LP}}$ with $b \in \{CR, AB, SAA-RSP\}$. Figure 3 shows the individual bounds' distribution per solution method as a histogram and the resulting normal distribution estimated based on the data. Note that we set $\zeta = 1$ for $UB^{SAA-RSP}$ to compare the generated bound with the other ones. First, the small range on the x-axis must be considered. Since the bounds for one solution method are all close together, the standard deviation is tiny in each case, i. e., with a minimum of 0.0014 for CR with one priority and at most 0.0025 for the SAA-RSP algorithm with one priority. A surprising result for the instances with only one priority (top) is that UB^{CR} is stronger than $UB^{SAA-RSP}$ with $\zeta = 1$. Remember that $UB^{SAA-RSP}$ is based on the lower bound of the deterministic model (see proposition 2) and for the deterministic instances we had – obviously – $UB^{CR} > LB^{SAA-RSP}$ (see Table 4). Since the CR for cases with two priorities is not solved within the time limit, the bound's quality substantially shifts. It is still better than the LP relaxation, but worse than the AB. Looking at the distance between the distribution of AB and the distribution of SAA-RSP, it can be seen that there is no difference between using one or two priorities, i. e., both with a mean difference of 0.06 ($\frac{UB^{AB}}{UB^{LP}} - \frac{UB^{SAA-RSP}}{UB^{LP}}$) and a p-value of 0.2810. Interestingly, this is similar to the distance between the CR

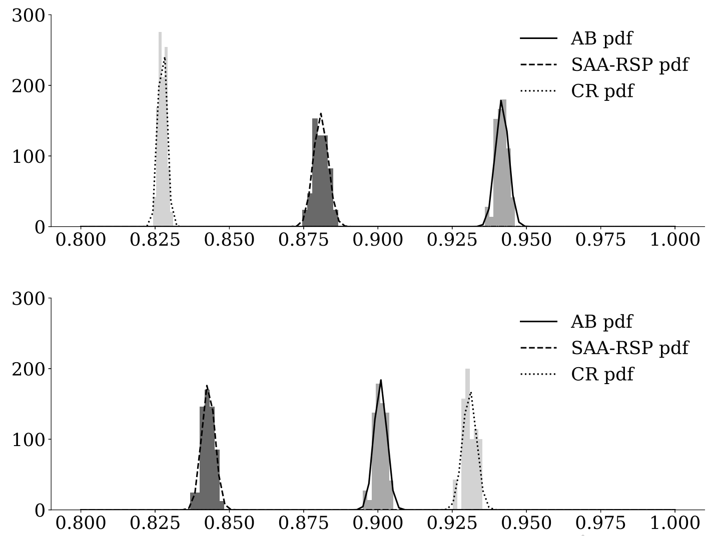

Figure 3: Probability density function and histogram of the sample of the upper bounds normalized by the LP relaxation (top: $|\mathscr{P}| = 1$, bottom: $|\mathscr{P}| = 2$)

distribution and the SAA-RSP distribution when the lower bound is used for CRs with two priorities (assuming the lower bound is the optimal solution), i. e., the distance between CR and SAA-RSP for one and two priorities with a mean difference of 0.05. Nevertheless, they are significantly different, with a p-value of $3.15 \cdot 10^{-25}$. Assuming that an upper bound is always in this interval, we can determine $UB^{SAA-RSP}$ with $\zeta = 0.95$. Note that ζ is derived from the lower bound of the CR, i. e., we might underestimate the upper bound. However, we do know that the AB always applies. Comparing the AB with our approximation a possible underestimation of up to 12% can be derived. In the performance analysis of the algorithm, we will consider the bound accordingly.

Performance of the SAA-RSP algorithm. In this part of the study, we will further analyze the performance of the SAA-RSP algorithm. For this purpose, we will examine three different variants in more detail—first, the (classical) SAA-RSP algorithm (Cl), as described in Section 4. Second, the same algorithm, but with a warm start (WS). Since we solve the deterministic problem to determine the upper bound, this solution's training schedule is used to initialize the algorithm in the first iteration. Finally, the batching method (see Section 4) is analyzed. Here we will randomly use $1/10$ of the scenarios in each iteration, i. e., $|\mathscr{B}| = \lfloor \frac{|\mathscr{S}|}{10} \rfloor$. For the batching, a warm start is used as well. As a termination criterion of the batching procedure, a total runtime of one hour is chosen. We will use the same 10 settings (5 instances) as in the first study. However, the number of scenarios ($|\mathscr{S}|$) is now set to 10 and 50. In a second step, we will analyze the performance of the scenario size (10 vs. 50) with an out of sample evaluation, i. e., we evaluate the performance of the generated training schedule with 500 different scenario trees. This is to show that the selected number of scenarios is sufficient for the problem.

The results of the first part of the study are given in Table 5. We do not separate the instances with 10 and 50 scenarios as there are no differences except the total runtime. The total runtime increases linearly with the number of scenarios. The first column of Table 5 shows the number of priorities. The second column identifies the algorithm used. Columns three to six refer to the performance indicators. Column 3 indicates the average optimality gap. Here it becomes clear that the SAA-RSP algorithm with two priorities leads to better results, i. e., an optimality gap of 7.24% with one priority and 4.91% with two priorities. However, this is accompanied by a longer runtime. The average number of iterations increases by 10, as shown in the fourth column. The same can be seen in the algorithms runtime for solving the second stage in parallel (Column 5) and not in parallel (Column 6).

Table 5: Average performance results of the SAA-RSP algorithm

$\|\mathscr{P}\|$	Algorithm	Gap (%)	Iterations	Total Time (s)	Aggregated Time (s)
1	SAA-RSP	7.24	7.4	158	4,288
	SAA-RSP (warm start)	3.72	1.0	64	1,657
	SAA-RSP (batching)	-	-	-	-
2	SAA-RSP	4.91	17.5	464	10,385
	SAA-RSP (warm start)	2.56	15.1	442	8,506
	SAA-RSP (batching)	2.56	35.6	3,350	3,350

It is interesting that the SAA-RSP with a warm start and $|\mathscr{P}| = 1$ only needs one iteration to terminate. This is mainly because of the randomness of absences, i. e., a resident is often only absent for one day. The initialized training schedule is already at a (local) optimum and cannot change further with only one priority. If the number of priorities is set to two, the number of iterations and the runtime increases significantly. On average, it also remains well below that of the classic SAA-RSP. The optimality gap is even better in each of the 100 instances, with an average of 2.56%.

Batching is only performed for instances with two priorities because the (local) optimum is also given in this approach for one priority. In most cases, the batching terminates due to the time limit. The average aggregated runtime is 3,350 seconds. However, because of the smaller number of simultaneously considered scenarios (10 per iteration), this method can perform significantly more iterations in time – on average, 35.6. A rather surprising result is the optimality gap. This gap averages 2.56%, like the warm start. One reason for the good performance of batching can be the change of scenarios. As the procedure often solves a different combination of scenarios, the algorithm does not necessarily move to a fixed local optimum. Concluding, batching is not only advantageous for reasons of computing capacity – the total runtime was fixed at one hour in the tests – but is also equivalent to a warm start in terms of solution quality.

To show that the chosen total number of scenarios (10 or 50) is sufficient for the evaluation, we test the generated solution of the training schedule on scenarios out of the sample, i. e., the second stage is run once for every scenario with the final training schedule of an instance as input (**?**). For this purpose, we use 500 additional scenario trees generated in the same way as before. The objective function value is not a good tool for comparison since it depends significantly on the number of missing residents. For this reason, we compare the number of residents with unexpected assignments, i. e., assignments that deviate from the training schedule. We follow a two-step approach. First, we analyze differences in the expected value. The results show that the deviations are significantly different with a p-value of $2.4 \cdot 10^{-24}$ when the training schedule is determined with only 10 scenarios, i. e., 10 scenarios are not sufficient to generate a stable solution outside the sample. In contrast, there is no significant difference with a p-value of 0.06 when the training schedule is determined with 50 scenarios. Second, we evaluate whether the variance of different scenario trees is the same. For the training schedule with 50 scenarios, we could not find a significant difference in the number of unexpected assignments using a Levene test with the smallest p-value of 0.05. Thus, we can conclude that a scenario size of 50 is sufficient to generate a stable training schedule for our problem.

5.2 Managerial insights of annual resident scheduling

In this part of the study, we will take a closer look at the managerial implications of our modeling approach. First, we analyze the effect of training priorities by assuming a different number of priorities for the same instances. In the second part of our analysis, we will change the number of overnight duties and the workforce size to evaluate their impact.

Effect of training priorities. In this last part of the study, we analyze the effect of priorities on the stability of the training schedule. Remember that a training schedule is stable if all assignments in the daily schedule match the assignments in the training schedule. We will use the average number of unexpected assignments as a measure for the stability. For this purpose, we use the previous studies' results and consider the daily schedule assignments in detail, i. e., the daily schedules of all used scenarios. Additionally, we extend the number of priorities to at most three and consider absence rates of 5%, 10%, and 20%. Thus almost 1.000,000 individual plans from the daily schedules are analyzed, i. e., all rosters from the scenarios of the different instances. As in the previous study, there is no difference in the number of scenarios for the instances, so in the following, we only distinguish between the number of priorities and the absence rate. Table 6 summarizes the assignments of each resident in the daily schedule according to the resident schedule. We first discuss the base case with an absence rate of 10% and then compare the results with the absence rate of 5% and 20%.

The first three columns declare each row's information, i. e., the average, minimum, maximum value using one, two, and three priorities with an absence rate of 5%, 10%, and 20%. The next column gives information about the average number of total assignments per resident. On average, each resident has 144.3 assignments in the daily schedule over the year. While the average number of assignments is identical regardless of the number of priorities, the maximum (from 177 to 180) and minimum (from 105 to 112) number changes depending on the number of priorities. Remember that a resident cannot be assigned to the daily schedule after an overnight duty. 90% of all assignments are correct according to the training schedule using only one priority. Consequently, an average of 10% of the assignments does not correspond to the training schedule. This leads to more than two months of assignments in an unexpected department (45 days) in individual cases. Since these assignments are not made in one go, this can lead to a loss of quality in the respective departments and the training of the residents. If the number of possible priorities is increased from one to two, the situation already changes drastically. Surprisingly, with two and three priorities, the algorithm can assign on average more residents with their first priority in the daily schedule, i. e., 93% versus 90%. One reason for this could be that the algorithm with two priorities drives to other solution regions and thus reaches different (local) optima. However, the most interesting point is the number of unexpected assignments. These can be almost completely eliminated with two priorities, i. e., on average, one unexpected assignment per resident in a year. Using three priorities reduces the number even further, with at most one unexpected assignment per resident. From this it can be concluded that in the case under consideration already three out of a maximum of 14 possible priorities are sufficient to generate a stable training schedule. If we compare

Table 6: Overview of the assignments in the daily schedule per resident according to the training schedule

			Total assignments	With $p = 1$	With $p = 2$	With $p = 3$	Unexpected assignments
$\lvert\mathscr{P}\rvert = 1$	5%	Mean	154.6	143.5	-	-	11.1
		Min	123	95	-	-	0
		Max	189	189	-	-	41
	10%	Mean	144.3	130.1	-	-	14.1
		Min	112	80	-	-	0
		Max	178	178	-	-	45
	20%	Mean	123.6	116.0	-	-	7.5
		Min	96	81	-	-	0
		Max	150	148	-	-	33
$\lvert\mathscr{P}\rvert = 2$	5%	Mean	154.6	143.4	9.4	-	1.8
		Min	120	94	0	-	0
		Max	194	191	31	-	17
	10%	Mean	144.3	134.4	8.8	-	1.1
		Min	105	81	0	-	0
		Max	180	178	35	-	14
	20%	Mean	123.6	116.1	7.0	-	0.4
		Min	96	80	0	-	0
		Max	153	150	30	-	8
$\lvert\mathscr{P}\rvert = 3$	5%	Mean	154.6	143.4	9.4	0.1	1.6
		Min	117	88	0	0	0
		Max	193	189	34	4	16
	10%	Mean	144.3	134.4	9.8	0.1	0.0
		Min	110	89	0	0	0
		Max	177	176	39	4	1
	20%	Mean	123.6	116.1	7.1	0.0	0.4
		Min	96	78	0	0	0
		Max	153	149	29	3	7

the different absence rates, we can see a similar behavior, i. e., two priorities support the planner with a stable training schedule. However, we can see an interesting aspect with one priority in the 20% case. The average number of unexpected assignments is lower than in the other two cases, i. e., 7.5 (20%) versus 11.1 (5%) and 14.1 (10%). This is because the number of absent residents is higher than the number of potential backups, i. e., some positions cannot be reassigned due to absences.

An advantage from an application perspective is that not every resident is assigned to a department in every period with all priorities, i. e., Constraints (2) allow a maximum of one assignment per priority. Consequently, residents do not receive unnecessarily more information, i. e., the model determines the potential replacement residents. Additionally, the planner has fewer options to choose from in case of absences. This is an advantage since the operational re-planning will not affect the training progress of the individual residents. Considering the number of priorities, we were able to show that just two priorities are sufficient to eliminate almost all unexpected assignments. While three priorities have slightly fewer unexpected assignments than two priorities, the information gain is out of balance in our view, i. e., a resident is assigned to a department with priority three at most four times over the year. Based on these results, we recommend at least two priorities in the annual planning of residents for our use case.

Effect of duties and workforce size. In Section 1, we motivated that overnight duties affect the training progress since a resident needs a day off after working an overnight duty. From a managerial perspective, hospital management can allow more experienced physicians (specialists) to work an overnight duty if it is not part of the training program. We analyze the impact of overnight duties on training. In addition, we analyze the impact of changing staffing levels on the stability of the training schedule. We analyze a total of eight different settings, i.e., we consider staffing levels of 64, 72, 80 (base case), and 160 residents with and without all 17 types of duties. The other parameters stay unchanged, except for the case with 160 residents. Here, we increase the minimum ($\underline{D}_{dt}^{\text{daily}}$) and maximum demand of the departments ($\overline{D}_{dt}^{\text{daily}}$) by 100% as well. We analyze the different cases in the same way as in the previous study, i. e., we measure the number of assignments according to the training schedule. The results are given in Table 7. The first two columns give information on the analyzed case, i. e., the number of

Table 7: Average assignments in the daily schedule per resident according to the training schedule

$\lvert I \rvert$	Duties	Total assignments	With $p = 1$	With $p = 2$	Unexpected assignments
64	on	133.6	129.2	4.4	0.0
	off	187.0	164.0	22.1	1.0
72	on	139.5	132.3	6.8	0.3
	off	187.1	157.3	21.3	0.8
80	on	144.3	134.4	8.8	1.1
	off	187.1	156.0	19.9	11.2
160	on	181.7	142.3	31.0	8.4
	off	203.0	164.4	27.2	11.5

residents $\lvert I \rvert$ and if duties are considered (on) or not (off). The next column gives information about the average number of total assignments per resident. An expected result is that residents can be assigned more often to a department if they do not have to work overnight duties (no day off requirement on the next working day), i. e., the training progress can be increased. Moreover, this effect is higher when the staffing level decreases. While the average number of assignments for the base case (80 residents) increases by 42.8 assignments from 144.3, this difference is 47.6 for 10% and 53.4 for 20% workforce reduction. On the other hand, if the number of residents increases by 100%, the effect is smaller in magnitude, with 21.3 additional assignments, but still there. These values precisely match the potential number of days off due to the overnight duties, which are not needed as duties are not considered. The last three columns show how many assignments match their first and second priority in the training schedule or are unexpected. A non-intuitive observation is that the average number of unexpected assignments increases when residents do not have to perform overnight duties. This is true in absolute values and relative to the total number of assignments. This behavior is especially prevalent in the case of 80 residents, where the number of unexpected assignments increases by 10.1 assignments per resident. One reason is that the different requirements, i. e., for duty and department demand as well as the training, are matched with the number of residents (i. e., the staff and requirements planning are well aligned in our real-world data). When duties no longer need to be staffed (by residents), more residents are in the system daily. As a consequence, the upper demand limit of departments ($\overline{D}_{dt}^{\text{daily}}$) is reached more often, so residents have to be assigned to another department. If this is the case, the number of priorities would have to be increased, as shown in the previous study, to avoid unexpected assignments. Please note that increasing the upper demand limit of a department cannot easily be done as the value, amongst other things, depends on the infrastructure (e. g., the number of operating rooms to provide service). Additionally, we can see that decreasing the number of residents increases the relative number of assignments according to the first priority in the training schedule. Reducing the number of residents by 20% increases the number of first-priority assignments by 4%. If the number of residents is increased by 100%, this value decreases by 14%. To summarize our findings, reducing the number of duties has a positive effect on the training of the residents, i. e., they can be assigned more often to a department necessary for training. However, the trade-off for the increase in total assignments is that more priorities are needed to reduce the number of unexpected assignments. Overall, we show that the system setting affects training and that some hidden relations exist. Therefore, management should be cautious when changing the setting (e. g., relieving residents from duty assignments).

6 Conclusions

The annual planning of residents is a complex and recurring problem. Although the problem is suitable for automated planning, it is still often done by hand. Since departments' changes for residents are accompanied by much preparation and stress, a stable training schedule is useful. Due to physicians' increasing shortage, hospitals are competing with each other to attract the best residents. Consequently, they have to emphasize themselves on different levels against their competitors. One possibility is to provide high-quality training schedules that ensure the defined goals are met in time.

In this work, we investigated the operative level's effects in terms of daily and duty scheduling on residents' annual planning in terms of planning reliability. We could identify a changing demand within a week, absences resulting from overnight duties, and other absences such as workshops, conferences or sickness as sources of disturbance. To avoid unforeseen changes in departments due to these disturbances, we developed an innovative new formulation for the annual planning of residents, which has a schedule with multiple assignments (priorities) per resident as output. By using different priorities on a tactical level, residents get more information about their planed assignments. The planners also get a smaller selection of possible replacements in case of absences without neglecting training goals. For short term re-planning, unplanned delays in training can be prevented. Besides, we have derived an analytic bound for the problem formulation, which is superior to the LP bound and can be calculated with almost no time effort. Furthermore, we have derived a bound by our solution method. Although this work deals specifically with the planning of residents, this model can also be applied to other areas where personnel can be assigned to different departments/ areas.

In the experimental study, we showed that the analytic bound becomes increasingly stronger as the number of priorities increases. The performance analysis of the SAA-RSP algorithm showed that a warm start with the solution of the deterministic problem and in combination with a batching scheme is advantageous both in terms of solution quality and solution time. When it comes to the number of priorities, the hospital management can choose between one and the number of departments. While with each additional priority the probability of an assignment corresponding to the training schedule increases, the increasing number of priorities decreases the quality of information, e. g., with 5 different departments and a training schedule with 5 different priorities it is clear that one will be assigned to one of these departments. For our case, we were able to show that just two out of at most 14 priorities, i. e., preferable sequence of the subset of departments, are sufficient to eliminate almost all unexpected assignments. While the appropriate number of priorities may vary depending on the case, the biggest reduction in unexpected assignments is provided by one additional priority. However, we were also able to identify the limitations of this approach. For example, a planner can only reassign positions if enough residents are available, i. e., there is a correlation between the minimum demand per department and the number of residents employed. Based on these results, we recommend at least two priorities in the annual training planning of residents. This allows minimizing re-planning efforts in terms of the training schedule and avoids possible deviations from training goals by residents. Consequently, our approach saves additional planning time and enables the residents to better prepare for the possible assignments. Eventually, this will increase the quality of the training and the quality of care in the long term. An interesting outcome is that reducing the number of duties increases the number of unexpected assignments. Meaning that hospital management has to deal with the trade-off between additional assignments relevant to the training and the number of priorities to generate a stable training schedule.

The literature review showed that resident scheduling problems often consider a single case. In order to give future researchers an easy entry into the topic, a provision of test data sets with different characteristics would be a desirable contribution. Even though the SAA-RSP algorithm delivers excellent results, the process can still be improved. It is not always possible to find an optimal solution in its current form because the procedure has no exploration step, i. e., it cannot escape from a local optimum. However, the literature has many ideas to avoid this problem. A simple possibility would be to use different feasible training schedules as a starting solution. Moreover, alternative solution techniques can be used to model the uncertainty of the problem and give some theoretical contributions. Instead of a two-stage solution approach, the dynamics of the problem could be considered using a multi-stage approach. This approach should be possible since the uncertainties are realized independently and incur at different times during the horizon. Herewith, it should be possible to follow the training program more accurately. A possible price is a delayed information transfer about the changes in the schedule. Identifying and evaluating this trade-off would fill an interesting research gap. Besides improving the solution procedure, one can also consider possible extensions. By decomposing the problem, both the tactical and operational levels can be extended almost independently. Especially the tactical level with the training schedule has potential because this level can be solved quickly. The operational level can be exchanged almost one-to-one with other models of operational planning from the existing literature. However, it must be considered that the operative models' solution is the procedure's bottleneck, and more complex models often increase the solution time.

Appendix

Please find a summary of all used notation in the following.

Sets with indices

$\mathscr{I}$	set of residents (index i)
$\mathscr{J}$	set of departments (index j)
$\mathscr{D}$	set of overnight duties (index d)
$\mathscr{W}$	set of weeks (index w)
$\mathscr{T}$	set of days in a week (index t)
$\mathscr{T}^{\text{work}}$	set of working days in a week ($\mathscr{T}^{work} \subseteq \mathscr{T}$)
$\mathscr{L}$	set of seniority level (index l)
$\mathscr{P}$	set of priorities (index p)
$\mathscr{S}$	set of scenarios (index s)

Parameters

T^{off}_{iwts}	1, if resident i is absent on day t in week w in scenario s, 0, otherwise
L^{duty}_{dl}	1, if overnight duty d requires seniority level l, 0, otherwise
L^{daily}_{jl}	1, if department j requires seniority level l, 0, otherwise
$L^{\text{resident}}_{ilw}$	1, if resident i has seniority level l in week w, 0, otherwise
G^{24h}	maximum number of overnight duties to be assigned to one resident in a single week
$\overline{D}^{\text{duty}}_{dt}$	maximum demand of residents for overnight duty d on day t
$\overline{D}^{\text{daily}}_{jt}$	maximum demand of residents for department j on day t
$\underline{D}^{\text{daily}}_{jt}$	minimum demand of residents for department j on day t
$\overline{M}^{\text{training}}_{ij}$	maximum number of weeks resident i should be assigned to department j in the training schedule
$\underline{M}^{\text{training}}_{ij}$	minimum number of weeks resident i should be assigned to department j in the training schedule
K_j	block length of department j in the training schedule
C^{daily}	cost per resident missing to satisfy minimum demand on a department
R^{duty}	reward for assigning a resident to an overnight or weekend duty
R^{daily}	reward for assigning a resident to a department
R^{training}_{p}	reward for assigning a resident to a department as planned in the training schedule
p_s	probability for a realization of scenario s

Decision variables

x_{ijwp}	1, if resident i is assigned to department j in week w with priority p in the training schedule, 0, otherwise
y_{ijwts}	1, if resident i is assigned to department j on day t in week w in scenario s, 0, otherwise
z_{idwts}	1, if resident i is assigned to duty d on day t in week w in scenario s, 0, otherwise
$\delta^{\text{daily}}_{jwts}$	resident deficit for department j on day t in scenario s in week w
π_{ijwpts}	1, if resident i is assigned to department j in week w with priority p in scenario s in the training schedule and daily schedule, 0, otherwise

$$\begin{aligned}\max \sum_{i\in\mathscr{I}}\sum_{j\in\mathscr{J}}\sum_{w\in\mathscr{W}}\sum_{t\in\mathscr{T}}\sum_{p\in\mathscr{P}}\sum_{s\in\mathscr{S}} R^{\text{training}}_{p} p_s \pi_{ijwpts}\\ +\sum_{i\in\mathscr{I}}\sum_{j\in\mathscr{J}}\sum_{w\in\mathscr{W}}\sum_{t\in\mathscr{T}}\sum_{s\in\mathscr{S}} R^{\text{daily}} p_s y_{ijwts} + \sum_{i\in\mathscr{I}}\sum_{d\in\mathscr{D}}\sum_{w\in\mathscr{W}}\sum_{t\in\mathscr{T}}\sum_{s\in\mathscr{S}} R^{\text{duty}} p_s z_{idwts}\\ -\sum_{j\in\mathscr{J}}\sum_{w\in\mathscr{W}}\sum_{t\in\mathscr{T}}\sum_{s\in\mathscr{S}} C^{\text{daily}} p_s \delta^{\text{daily}}_{jwts}\end{aligned}$$

$$\sum_{j\in\mathscr{J}} x_{ijwp} \le 1 \qquad \forall i\in\mathscr{I}, w\in\mathscr{W}, p\in\mathscr{P}$$

$$\sum_{w\in\mathscr{W}}\sum_{p\in\mathscr{P}} x_{ijwp} \le \overline{M}^{\text{training}}_{ij} \qquad \forall i\in\mathscr{I}, j\in\mathscr{J}$$

$$\sum_{w\in\mathscr{W}} x_{ijw1} \ge \underline{M}^{\text{training}}_{ij} \qquad \forall i\in\mathscr{I}, j\in\mathscr{J}$$

$$\sum_{j\in\mathscr{J}} x_{ijwp} - \sum_{j\in\mathscr{J}} x_{ijwp'} \ge 0 \qquad \forall i\in\mathscr{I}, w\in\mathscr{W}, p,p'\in\mathscr{P}, p<p'$$

$$x_{ijwp} + x_{ijwp'} \leq 1 \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, w \in \mathscr{W}, p, p' \in \mathscr{P}, p < p'$$

$$x_{ijw'p} \geq x_{ijwp} - x_{ijw-1p} \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, w \in \mathscr{W} \setminus \{|\mathscr{W}| - K_j + 1, \dots, |\mathscr{W}|\}, p \in \mathscr{P}, w' \in \{w+1, \dots, w+K_j - 1\}$$

$$x_{ijw'p} \geq x_{ij1p} \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, p \in \mathscr{P}, w' \in \{2, \dots, K_j\}$$

$$\sum_{i \in \mathscr{I}} y_{ijwts} \leq \overline{D}^{\text{daily}}_{jt} \qquad \forall j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T}, s \in \mathscr{S}$$

$$\sum_{i \in \mathscr{I}} y_{ijwts} + \delta^{\text{daily}}_{jwts} \geq \underline{D}^{\text{daily}}_{jt} \qquad \forall j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T}, s \in \mathscr{S}$$

$$\sum_{j \in \mathscr{J}} y_{ijwts} \leq 1 - T^{\text{off}}_{iwts} \qquad \forall i \in \mathscr{I}, w \in \mathscr{W}, t \in \mathscr{T}, s \in \mathscr{S}$$

$$y_{ijwts} \leq \sum_{l \in \mathscr{L}} (L^{\text{daily}}_{jl} L^{\text{resident}}_{ilw}) \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T}, s \in \mathscr{S}$$

$$\sum_{i \in \mathscr{I}} z_{idwts} \leq \overline{D}^{\text{duty}}_{dt} \qquad \forall d \in \mathscr{D}, w \in \mathscr{W}, t \in \mathscr{T}, s \in \mathscr{S}$$

$$\sum_{d \in \mathscr{D}} z_{idwts} \leq 1 - T^{\text{off}}_{iwts} \qquad \forall i \in \mathscr{I}, w \in \mathscr{W}, t \in \mathscr{T}, s \in \mathscr{S}$$

$$z_{idwts} \leq \sum_{l \in \mathscr{L}} (L^{\text{duty}}_{dl} L^{\text{resident}}_{ilw}) \qquad \forall i \in \mathscr{I}, d \in \mathscr{D}, w \in \mathscr{W}, t \in \mathscr{T}, s \in \mathscr{S}$$

$$\sum_{d \in \mathscr{D}} \sum_{t \in \mathscr{T}} z_{idwts} \leq G^{24\text{h}} \qquad \forall i \in \mathscr{I}, w \in \mathscr{W}, s \in \mathscr{S}$$

$$\sum_{d \in \mathscr{D}} 3 z_{idw(t-1)s} \leq 3 - \sum_{d \in \mathscr{D}} z_{idwts} - \sum_{j \in \mathscr{J}} y_{ijwts} - T^{\text{off}}_{iwts} \qquad \forall i \in \mathscr{I}, w \in \mathscr{W}, t \in \mathscr{T}, s \in \mathscr{S}, t-1 > 0$$

$$\sum_{d \in \mathscr{D}} 3 z_{id(w-1)7s} \leq 3 - \sum_{d \in \mathscr{D}} z_{idw1s} - \sum_{j \in \mathscr{J}} y_{ijw1s} - T^{\text{off}}_{iw1s} \qquad \forall i \in \mathscr{I}, w \in \mathscr{W}, s \in \mathscr{S}, w-1 > 0$$

$$\pi_{ijwtps} \leq x_{ijwp} \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T}, p \in \mathscr{P}, s \in \mathscr{S}$$

$$\pi_{ijwtps} \leq y_{ijwts} \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T}, p \in \mathscr{P}, s \in \mathscr{S}$$

$$\pi_{ijwtps} \geq x_{ijwp} + y_{ijwts} - 1 \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T}, p \in \mathscr{P}, s \in \mathscr{S}$$

$$x_{ijwp}, y_{ijwts}, z_{idwts} \in \{0, 1\} \qquad \forall i \in \mathscr{I}, j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T}, p \in \mathscr{P}, s \in \mathscr{S}$$

$$\delta^{\text{daily}}_{jwts} \in \mathbb{Z}_+ \qquad \forall j \in \mathscr{J}, w \in \mathscr{W}, t \in \mathscr{T}, s \in \mathscr{S}$$