Technical University of Denmark
Department of Technology, Management, and Economics Management Science

Physician Scheduling at Rigshospitalet

a Case Study of the Anesthesia Department

Master Thesis

8	9	10	11	12	13	14	15	16	17	18	19	20	21	2
Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
FF	FF	FF	FRI	FRI	BF	SOV	FRI	F1	ADM	AKC	B2	SOV	SEMI+L	ADM
SME	D+R	SOV	FRI	FRI	FRI	SME	SME	FRI	FRI	FRI	FRI	SME	TJU	SME
SEMI	TC	FRI	FRI	FRI	TC	TC	TC	TC+AKC	SEMI	FRI	B1	D8	TC	TC+AKC
B	SOV	FRI	AKC	B2	SOV	D+AKC	SEMI	FRI	TR+L	FRI	FRI	TR+L	ALD	D17
D8	D	HE	FRI	BF1	FRI	FRI	FRI	HE	HE	FRI	FRI	ADM	HE	ADM
SOV	SEMI	SEMI+L	BF1	FRI	BLO	FRI	FRI	BLO	BLO	FRI	FRI	BLO	D17	BLO
D	D	D	FRI	wD+R	ADM	D	ADM	D	D	FRI	FRI	D18	D	SEMI+L
HEMS	FRI	FRI	FRI	FRI	HEMS+L	HEMS	HEMS+R	SOV	D+AKC	FRI	FRI	TJ	TJ	TJ
FE	FE	FE	FE	FE	D+AKC	B	SOV	SEMI+L	FE	FE	FE	FE	FE	ALD
D	D17	D	FRI	FRI	D	D17	D	D	D	FRI	FRI	D	D	D
FRI	FRI	FRI	FRI	FRI	SEMI+L	F1	D8	D	DLF	BF2	SOV	F1	DLF	D
FRI	FRI	FRI	FRI	B1	D	SEMI+L	B	SOV	D8	FRI	FRI	SEMI+L	D	DLF
SOV	D	OM5	FRI	FRI	D8	BF	SOV	D	OM6	FRI	FRI	D	B	SOV
D	D	D	FRI	FRI	D18	D	D	D	D	FRI	FRI	BF	SOV	D+R
D17	ALN	SOV	FRI	FRI	D	D	D	D	FRI	FRI	FRI	D18	D	D
D17	-	LL	FRI	FRI	-	-	-	-	-	FRI	FRI	-	-	-
D	ADM	FRI	FRI	FRI	FRI	D	D	ADM	D	FRI	FRI	D	D+L	D
FRI	FRI	FRI	FRI	FRI	FRI	D	D	D	D	FRI	FRI	D	D	D
CA	CA	CA	FRI	FRI	PKL8	PKL9	CA	ALN	SOV	FRI	FRI	PKL10	D17	PKL11
D+L	D	FRI	FRI	FRI	FE	HBO	HBO	HBO	HBO	FRI	FRI	ADM	HBO	HBO
FRI	FRI	AKUT+AKC	B2	SOV	FE	FE	FE	FE	FE	FE	FE	D+AKC	D8	FRI
D+AKC	D	D+L	FRI	AKC	TJ	TJ	D	D8	D	FRI	FRI	TJ	D+AKC	TJ
ADM	B	SOV	FRI	FRI	D	DLF	ADM	FRI	B	SOV	BF2	SOV	D	ADM
TJU	TJU	TJU	FRI	FRI	D	ALD	ALN	SOV	D	FRI	FRI	D	D+R	SOV
FRI	FRI	FRI	FRI	FRI	D	D	D	D17	BF	SOV	FRI	TR+L	BF	SOV
D	D	BF	SOV	FRI	D18	D	BF	SOV	D	FRI	FRI	FE	FE	FE
ALD	D+AKC	D	FRI	FRI	D	D	D+AKC	D	D	FRI	AKC	ALN	SOV	D17
ALN	SOV	ALD	FRI	FRI	ALN	SOV	UFU	UFU	UFU	FRI	FRI	UFU	UFU	UFU
SOV	D+L	D	FRI	FRI	D	D+R	SOV	D+L	D	FRI	FRI	D	D+R	SOV
TJ	D	D	FRI	FRI	FE	FE	FE	FE	FE	FE	FE	DLF	TC	TC8
FRI	D	FO	FRI	FRI	D+L	FRI	D+L	D	FO	FRI	FRI	D+L	D	D
TJ	TJ	TJ	B1	FRI	B	SOV	ALD	FRI	D+L	FRI	FRI	FRI	FRI	B
D	BF	SOV	FRI	FRI	D	D8	D17	D	D	FRI	FRI	D	D	D
D	TJ	B	SOV	FRI	TC	D17	TC	ALD	ALD	FRI	wD+R	SOV	FRI	ALN
HBO	HBO	HBO	FRI	FRI	FO	FO	FO	FO	FO	FRI	FRI	HBO	FO	FO
FRI	DIG	FRI	BF2	SOV	DIG+R	DIG	TJ	ADM	F1	B2	SOV	ADM	FRI	DIG
SOV	TC8	TC8	FRI	FRI	TJU	TJU	TJU	TJU	TJU	FRI	FRI	TJU	TJU	TJU
D	ALD	D+R	SOV	BF2	SOV	D+L	D	D17	FRI	FRI	FRI	D	D	D
D	D17	D	FRI	FRI	FRI	D	D	D	FO+R	SOV	FRI	D+R	FO	D
D+R	ADM	D	FRI	FRI	FRI	ADM	D17	BF	SOV	wD+R	SOV	FRI	ADM	BF
FRI	D	D	FRI	FRI	ALD	ALN	SOV	DLF	D	BF1	FRI	ALD	ALN	SOV
BF	SOV	D	wD+R	SOV	HBO	FRI	D	D+R	SOV	FRI	FRI	D	TJU	D+L
TJ	TJ	TJ	FRI	FRI	D	D	DLF	B	SOV	B1	FRI	B	SOV	D
DLF	D	D	FRI	FRI	LL	D	LL	LL	LL	FRI	FRI	LL	LL	LL
LL	LL	D	FRI	FRI	DLF	D	LL	D	D	FRI	FRI	LL	D	D
LL	LL	DLF	FRI	FRI	TJ	TJ	TJ	LL	D	FRI	BF1	D	D	LL
FE	FRI	D	FRI	FRI	D	LL	D	D	D	FRI	FRI	D	D	D
TJ	DLF	LL	FRI	FRI	D	LL	D	D	LL	FRI	FRI	D	D	LL
LL	LL	LL	FRI	FRI	LL	LL	LL	D	FRI	FRI	FRI	D	LL	D

Physician Scheduling at Rigshospitalet
a Case Study of the Anesthesia Department

Master Thesis
July 22nd, 2021

By
Mathias Klinkby
Siv Sørensen

Cover photo: Schedule generated by the developed model

Published by: DTU, Department of Technology, Management, and Economics Management Science, Akademivej, Building 358, 2800 Kgs. Lyngby Denmark
www.man.dtu.dk

Preface

This thesis has been prepared over a six months period at the Section for Operations Research, Department of Technology, Management, and Economics Management Science, at the Technical University of Denmark, DTU, in partial fulfilment for the Master of Science in Engineering degree, MSc Eng.

The contribution of each author is equivalent to 30ECTS. Both authors have contributed equally to all aspects of the thesis and take full responsibility for the entirety of the works submitted.

Mathias Klinkby - s164339

Signature

7/22/2021

Date

Siv Sørensen - s164374

Signature

7/22/2021

Date

Abstract

The process of generating rosters for physicians is a task most commonly performed by a chief physician in each department. The task is mostly carried out manually despite many hospitals investing in various decision supporting tools. There is a consensus among the appointed chief physician schedulers that the task of creating rosters is both unpleasant and time consuming.
In this thesis a MIP model is developed which generates monthly rosters for the anesthesia department at Rigshospitalet, Denmark. The model primarily focuses on satisfying non-financial targets through a weighted sum approach. The model has been developed with the aim of generating rosters of high practical quality, thus, alleviating the manual work for the scheduler. Besides common scheduling elements, the model also includes split employment contracts, stand-by services, Danish work time regulations, employee wishes, personal targets and preferences, and fairness. The fairness aspects included are fairness on the amount of over hours, fairness on the fraction of personal wishes granted, and equal distribution of unpleasant shifts.

The model is initially solved by a standard MIP formulation approach, using the Gurobi 9.1.1 MIP solver. To improve performance, a lexicographic approach as well as a Dantzig-Wolfe decomposition approach are also tested. We conclude that a warm start approach combined with the MIP formulation approach provides the best solution performance. This approach produces solution with an optimally gap of $\sim 15\%$ within 12 hours.
Despite the poor theoretical bounds achieved, the practical quality of the generated rosters are evaluated to be up to 90% optimal from the scheduler's perspective. The evaluation of the scheduler is based on a real life implementation of three rosters in the anesthesia department.

Keywords: Physician rostering, Mixed Integer Programming, MIP, Weighted sum objective function, Lexicographic optimization, Decomposition

Acknowledgements

Mathias Klinkby, Industrial Engineering and Management Engineering, DTU
Siv Sørensen, Mathematical Modelling and Computation Engineering, DTU

First and foremost, we are grateful for the Technical University of Denmark for realizing this thesis and for all of the academic opportunities given to us throughout our five years of studies. Furthermore, we wish to thank our thesis supervisor **Thomas Riis Stidsen**, Associate Professor at DTU, for all of his support, guidance, and help. Thomas was also the one, who at our request, found this project through his contact, Christian Sørup, at Rigshospitalet. For this we are grateful.

We would also like to extend our gratitude to both **Christian Sørup**, Decision Scientist, and **Rasmus Winkel**, Chief Physician, for their time and dedication to this project. Christian has through his current work at the Unit of Data and Quality at Rigshospitalet and his background at DTU, facilitated and initiated fruitful dialogues between the two parties, DTU and Rigshospitalet. Christian's network and reach at the more strategic level at Rigshospitalet has also been valuable to the project. As one of the most senior schedulers at Rigshospitalet, Rasmus has provided essential knowledge into the problem of creating a quality rooster for the anesthesia department at Rigshospitalet. Not only has Rasmus provided all of the data used in this thesis, he has also provided essential constructive feedback on the model limitations and results throughout the entire project.

Lastly, we would like to thank the Management department at DTU for allowing us unlimited access to their cluster. The cluster has been essential to the quality of the various analyses carried out as part of this thesis.

Contents

1 Introduction

An extensive amount of research has been carried out in the field of personnel scheduling problems since the 1950s [Petrovic, 2019]. We refer to [Ernst et al., 2004a] for an annotated bibliography of more than 700 papers organized by the type of scheduling problem, the application areas covered, and the methods used. The personnel scheduling problem, also commonly referred to as the rostering problem, can be defined as the process of constructing work timetables for the staff within an organization, such that the organization can satisfy its demand for goods as well as services [Ernst et al., 2004b]. Although the problem is easy to define, finding optimal or even good solutions is difficult. Personnel scheduling problem are heavily constrained and multi-objective problems, for which solutions should minimise costs, promote employee satisfaction and fairness, and live up to consumer needs and requirements, while also complying with industry regulations and local workplace agreements.

This study will focus on the personnel scheduling problem of generating a one month roster for the physicians at the anesthesia department at Rigshospitalet in Copenhagen, Denmark. The aim will be to generate rosters which require minimal adjustments and effort from the scheduler, while also aiming to develop a scalable model that could potentially be extended to other departments or hospitals in Denmark. However, within the scope of this thesis, the scalability of the model will not be tested. Here, the mathematical programming technique Mixed Integer Programming (MIP) will be applied to the problem. In recent years, MIP models have gained significant interest and popularity in the field of operations research. This is due to the flexible nature of the models and the continuous improvements in both computational hardware as well as the MIP algorithms used in commercials solvers [Bixby and Rothberg, 2007].

The first part of the study is an introduction to existing research within both scheduling in the healthcare sector and scheduling of physicians, specifically. The second part is an in depth presentation of the problem at hand, including a thorough walk-through of the data that defines the problem. In the third part, first, a base model is described, then a MVP model which is an extension of the base model, and finally, the full model, titled the Fairness model, which is an extension of the MVP model. In the fourth part, two alternative solution approaches to the full fairness model are defined; a lexicographical approach and a decomposition approach. The sixth part of the study presents the analyses and results of the various models and solution approaches. Lastly, a discussion and conclusion of the work carried out in the study as well as a discussion of future work is presented.

1.1 Literature study

Within the healthcare sector we distinguish between two general types of optimization problems: personnel problems and resource problems. In terms of their planning horizons, healthcare personnel problems can be further categorized into three sub-problems: *staffing*, *rostering*, and *re-planning* [Erhard et al., 2018]. *Staffing* problems are strategic long-term decision problems which concern the size and qualifications needed within a workforce. *Rostering* problems are mid-term decision problems for which the outcome is a concrete schedule or roster for a group of staff typically belonging to the same unit or department. These schedules typically span anywhere from one week to a few months and details daily shift assignments for each staff member. This project belongs to the *rostering* category.

Lastly, *re-planning* problems deal with the short-term schedule adjustments needed to handle staff illness, demand fluctuations, etc.

Healthcare resource problems concern the physical resources at a healthcare institution. Here, three of the most common and well researched problems are the operating theatre scheduling problem, the patient admission scheduling problem (PASP) and the outpatient scheduling problem. The operating theatre scheduling problem concerns the scheduling of all of the resources required for surgeries [Guerriero and Guido, 2011]. This includes both facility requirements such as operating rooms, recovery rooms, intensive care units, etc., as well as personnel requirements such as surgeons, nurses, anaesthetists, etc. The PASP, originally formalized by Demeester et al. [2010], consists of the allocation of patients to hospital beds which meet both equipment and location requirements while minimizing costs and bed transfers [Ceschia and Schaerf, 2012]. Lastly, the outpatient scheduling problem deals with the scheduling of all patients who do not need to stay overnight. Typically, the goal is to minimize waiting time for both patients as well as staff and optimize the use of medical equipment and staff resources while taking uncertainties such as treatment time, emergencies, no-shows, etc. into account [Cayirli and Veral, 2009].

For an overview of operations research done within the healthcare sector we refer to Rais and Viana [2011] and to Abdalkareem et al. [2021] who provide a more recent review of 190 papers published between 2010-2020. Although many optimization problems within the healthcare sector have been researched, multiple papers point out the flaw of tackling isolated problems that do not consider the healthcare institution as an integrated unit. Vanberkel et al. [2009] argued that management in healthcare institutions focus on the optimization of individual departments, failing to accurately account for their interactions. This comes at the expense of the total care chain from admission to discharge of patients which typically spans multiple departments. ”Not surprisingly, this has often resulted in diminished patient access without any significant reduction in costs” concludes de Bruin et al. [2006].

Another aspect of the healthcare sector which makes planning and resource optimization challenging is the inherent stochastic processes deeply rooted throughout the entire sector [Brunner et al., 2009]. The demand for both staff and medical resources is almost always stochastic in nature but is treated as deterministic in the majority of research. Erhard et al. [2018] found that 85% of all papers on physician scheduling up until 2016 adopt a deterministic approach. There are multiple factors of uncertainty which affect demand profiles such as the uncertainty related to emergencies [El-Rifai et al., 2015] and the duration of surgeries [Saadouli et al., 2015; Min and Yih, 2010]. Green et al. [2006] applied queuing theory to the process of scheduling staff in emergency departments while aiming to limit the fraction of patients who show up without being treated. Lamirim et al. [2009] suggested an ”almost” exact method combining Monte Carlo simulation and mixed integer programming for the stochastic problem of operating room sharing between elective and emergency surgeries. Here, elective surgeries are pre-scheduled while emergency patients arrive at random. Lamirim et al. [2009] also considered heuristic and meta-heuristic solution methods and found the taboo search method to be the best of the heuristic approaches considered for the stochastic problem.

When considering personnel optimization problems within the healthcare sector, the majority of research has focused on nurse scheduling problems, specifically the *rostering* problem [Brunner et al., 2009]. Burke et al. [2004] and Cheang et al. [2003] both present extensive reviews on the literature on nurse scheduling. Despite the apparent similarities between physician and nurse scheduling, the physician scheduling problem is renowned for

being more complex and less generic which arguably is why it has been studied less than the nurse scheduling problem. It is not uncommon that physicians have individual contract clauses, split employment between multiple departments, complex labor agreements, an extensive set of local agreements (which in extraordinary situations allow violations of the labor agreements), irregular shift definitions where nursing shifts typically begin and end at the same two-three points in time during a day, and a series of additional on-call shifts during off-hours and weekends [Brunner et al., 2009]. Not surprisingly, the scheduling problem, and more specifically a restricted version of the manpower shift scheduling problem have been proven to be $\mathcal{NP}$-complete by a reduction from 3SAT by Ullman [1975] and Lau [1996], respectively.
In an extensive review of the literature on physician scheduling between 1985-2016 by Erhard et al. [2018], only two of the 68 reviewed papers focus on developing a generic model with the aim of enabling adaption for different departments and hospitals. Rousseau et al. [2002] introduced a model with generic constraints enabling small amounts of customization in an attempt to handle the instance variations between departments and hospitals. They introduced generic *distribution constraints* which would allow the scheduler to control the number of shifts, weekends, nights, etc. worked for each physician during a set of chosen days using generic $<,=,>$ operators. They also introduced generic *pattern constraints* which the scheduler could use to model simple patterns of shifts that should be either forbidden or imposed. To the best of our knowledge, no generic model currently exists which successfully captures even just the portfolio of hard constraints within hospitals around the world, a view shared by Brunner et al. [2009]. Here, extensive, yet crucial, country-specific labor agreements presents one of the most challenging obstacles for generic model approaches.

Today, physician scheduling is still commonly carried out manually in each individual department by a chief physician [Brunner et al., 2009]. At the Ottawa Hospital Clinical Teaching Unit in Canada, a poll of former chief physicians reported that ”the single most unpleasant duty they faced was the creation of these schedules by hand and the reaction of the staff when they were posted” [White and White, 2003]. The scheduler and chief physician whom we have collaborated with during this project, has independently of the paper by White and White [2003] stated word by word almost the exact same on behalf of the schedulers at Rigshospitalet in Denmark. The backslash from staff commonly arises from what they perceive as lack of fairness in the schedule, whether it being related to ungranted wishes, an excess of certain shift in comparison to other staff members, or the general belief that other staff member has been assigned a ”better” roster.
The objectives in the physician scheduling problem are either financial or non-financial [Erhard et al., 2018], where fairness aspects are categorized as the latter. In general, past physician scheduling literature has focused on non-financial rather than financial goals [Erhard et al., 2018]. In the physician scheduling review by Erhard et al. [2018], four types of fairness aspects are singled out: Granted requests, even distribution of unpopular shifts, joint weekends, and equal distribution of free weekends. Non-financial goals are generally implemented as soft constraints and consequently penalized by minimizing the weighted sum of constraint violations in a multi-objective context. Despite criticism from both researchers and practitioners [Böðvarsdóttir et al., 2021], the weighted sum approach is commonly adapted in the literature as a way of handling the multi-objective nature of personnel scheduling problems. Arguments against the approach include both the imbalance between different objectives, the trade-offs that can be made between any two objectives both financial and non-financial, and the sensitivity towards the chosen ’weights’ for each of the objectives [Branke et al., 2008; Böðvarsdóttir et al., 2021].
Erhard et al. [2018] identified six papers which applied Goal Programming [Jones and

Tamiz, 2010] to the multi-objective formulations. New approaches for handling multi-objectivity has also been explored such as Behind-the-Scenes Weight Tuning, introduced by Böðvarsdóttir et al. [2021] for the application of nurse rostering, which uses measurable targets for guiding the process of automatically setting objective weights.

Lastly, we will briefly touch upon the various solution procedures used in the literature of physician scheduling problems. Exact solution approaches has dominated the field since around 2013 with Mixed Integer Programming (MIP) being the dominating modeling approach followed by Integer Programming (IP) [Erhard et al., 2018]. This could be due to the progress in the field of computational mixed integer programming as pointed out by Bixby and Rothberg [2007]. Topaloglu and Ozkarahan [2011] proposed a MIP formulation for the medical resident scheduling problem, considering both regulations related to on-call night duty, days off, rest periods, and total work-hours along with enforcing equal workload for each resident as an attempt to ensure fairness. The model was initially solved using a Branch and Cut algorithm but by decomposing the problem, Topaloglu and Ozkarahan [2011] managed to get an impressive reduction in solution time. de Kreuk et al. [2004] proposed a simulated annealing heuristic for scheduling the various tasks performed by medical specialists in a Dutch hospital. More recently, Cildoz et al. [2021] developed a GRASP-based algorithm for solving the emergency room physician scheduling problem in Spain for a one year planning period. The model included a variety of aspects such as demand, workload, ergonomics, fairness, etc. and is now in use. Cildoz et al. [2021] initially modelled the problem as an Integer Linear Programming (ILP) problem but found it was unsolvable using CPLEX with a time limit of one week. Subsequently, they developed a solution approach in which a GRASP construction heuristic provides an initial schedule, which is subsequently improved through a Variable Neighborhood Descent (VNDS) type algorithm, in combination with Network Flow Optimization (NFO) models.

1.2 The man-power planning problem at Rigshospitalet

The problem of designing a schedule for the physicians at the anesthesia department at Rigshospitalet contains the classic characteristics of a timetabling problem. In its essence, the solution to the problem is a schedule consisting of a set of assignments of shifts to every physician on a monthly basis. A schedule is feasible if it meets the demand for every shift type, correctly assigns every shift to a physician qualified to carry out the shift, and is compliant with the Danish Working Time Directive.
An example of a schedule for two physicians for the first two weeks of a month can be seen in Table 1.1.

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
CSV	FRI	TC+AKC	TC	TC	TC+AKC	B1	B1	FRI	TC	D8	SEMI	TC	FE	FE
JHØ	SOV	ALN	SOV	SEMI	D+L	FRI	FRI	D	D+AKC	SEMI	D8	ALD	FE	FE

Table 1.1: An example of a schedule for two physicians, CSV and JHØ, for the first two weeks of a month.

Each physician is assigned exactly one regular shift every day of the week, thus, possible shift assignments also include; `FRI` (free day), `FE` (vacation day), etc. Furthermore, every physician is also assigned exactly one add-on shift every day, on top of their regular shift. Examples of current add-on shifts are: on-call (stand-by) shifts for handling off-hour emergencies, shift extensions (long shifts) which are extensions of regular shifts, and a 'no add-on' option. A regular shift and an add-on shift are separated by a '+' in the plan e.g. `D+R` is a clinical day shift `D` and on-call add-on shift `R`, and `FO+AKC` is a research shift `FO` and on-call add-on shift `AKC`. For simplicity, 'no add-on' assignments are left out of the visual representation of a roster.

Today, each department at Rigshospitalet has a scheduler which manually creates the roster for each month. The rosters are typically created in Excel. The scheduler at the anesthesia department uses two full work days on average on creating a monthly roster. This time is spend on both gathering the wishes from the physicians, creating an initial roster, and sending the roster out for review, before finalizing it.

The extensive details of the man-power planning problem at Rigshospitalet are outlined in the presentation of the data below, which defines the boundaries of the problem. Thus, to get a thorough understanding of the problem, one must understand the underlying data.

1.3 Data introduction

For the purpose of this thesis, the data used to set up the model is structured into 13 sheets in Excel. Ideally, the data would be saved using appropriate data structures. This would allow the user (the scheduler) to edit, add, or delete data in a controlled and user friendly manner, as well as allowing the model to access and use the data most efficiently. However, the user interface and underlying data structures are considered beyond the scope of this thesis.

Presented below is a short summary on the content of each of the 13 data sheets. By understanding the data underlying the mathematical model, the reader will be able to follow and understand the construction of the model better. In the visual representations of the data sheets pictured in the sections below, a blank cell represents a zero.

1.3.1 Data sheet 1: Personnel

This sheet contains information about the seniority level, competencies, and affiliations of each physician.

Physician	STAT	NR	$Comp_1$	$Comp_2$	...	$Comp_{25}$
CSV		1	1	1		
...			1	1	1	
HJØ		1		1		1

Table 1.2: Simplified overview of the data in data sheet 1: Personnel.

STAT 1 if a physician is inactive, otherwise zero. An inactive physician will be excluded from the schedule generated by the model. The physician status parameter can be used for prolonged sick leave, maternity leave, etc.

NR 1 if a physician is a normal resource (NR), otherwise zero. A physician who is *not* a normal resource will only be assigned shifts when it is otherwise impossible to generate a feasible or descent schedule.

Comp$_x$ is 1 if a physician has the competence x, where $x = 1, ..., 25$, otherwise zero. Examples of competences include:

- Seniority level, i.e. chief physician (OLG), senior registrar (AFD), physician trainee (RLG), and specialist (OLG or AFD)[1].
- General shift competences, e.g ALB (ability to man medical car), HBO (ability to man pressure chamber), etc.
- Helper competences, such as a 'no competence' competence which is required for shifts with no prerequisites, a 'night' competence - the ability to work during the night, etc.
- Affiliations, such as primary floor level (1 or 2) and unions (FAS or YL).

1.3.2 Data sheet 2: Shift types

This sheet contains information about all shifts, both regular and add-ons. There are approximately 50 active regular shifts and six add-on shifts in the anesthesia department.

G1: Status 1 if a shift/add-on is *inactive*, otherwise zero. An inactive shift/add-on will be excluded from the schedule generated by the model.

G2: Rule type only applicable to shifts/add-ons outside normal working hours (7-17). Either tagged as 'presence (p)', 'on-call (o)', or 'alert (a)'. The unions have different rules for the different shift tags, e.g. there is a maximum number of 'presence (p)' shifts, which can be assigned to a physician over a norm period.

G3: Consecutive days 1 or 0. The shifts/add-ons assigned a 1 make up a set, where all pairwise shifts/add-ons in the set should not be assigned on consecutive days, preferably. E.g. a physician should not have two long shifts in a row.

G4: Wished 1 for shifts which can only be assigned to a physician who specifically wishes for that shift, otherwise 0. An example is vacation days: FE_h and FE_w.

[1] Translations acquired from: https://aalborguh.rn.dk/forskning/faa-stoette-til-din-forskning/kommunikation/formidling-paa-engelsk

	General													Week day compat.							Add-ons		
	Name:	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	Mon	Tue	Wed	Thu	Fri	Sat	Sun	None	...	AKC
Add-ons	None												0	1	1	1	1	1	1	1			
	R_OLG		o	1								09:00	4	1	1	1	1	1	1	1			
	R_AFD		o	1								09:00	0	1	1	1	1	1	1	1			
	L_475			1									4.75	1	1	1	1	1	1	1			
	L_0			1									0	1	1	1	1	1	1	1			
	AKC		a	1								07:00	0	1	1	1	1	1	1	1			
Shifts	D									1	07:45	15:15	7.5	1	1	1	1	1			1	1	1
	...																				1		
	FE_h				1		1	1			07:45	15:15	7.5	1	1	1	1	1			1		
	FE_w				1			1	1		07:45	15:15	0						1	1	1		

Table 1.3: Simplified overview of the data in data sheet 2: Shift types.

G5: Administrative 1 for shifts which can follow an 'on-call (o)' or 'alert (a)' night shift, otherwise 0. This includes only administrative shifts. Since a physician will not necessarily be contacted during an on-call or alert shifts, it is possible to work on administrative task the following day. Administrative tasks can be done in the evening if the physician ends up having to work during the night. This set of shifts can only be assigned after an 'on-call (o)' or 'alert (a)' night shift if a physicians has have agreed to it.

G6: Other shifts 1 if the working hours associated with the shift, do not count towards either the hours working in the clinic or the hours working of a separate contract. E.g. vacation shifts, sick days, paid and unpaid duty free days, etc.

G7: Contract dependency 1 if the working hours associated with the shift are directly correlated with the employment percentage of the physician assigned to the shift, otherwise zero. E.g. for a full time physician, a vacation day on Mon-Fri, FE_h, counts as 7.5 hours, whereas a physician on a 80% of full time contract only receives $7.5 \cdot 0.8$ hours for the same FE_h 'shift'.

G8: Rest period 1 if shift is part of a rest period (fridøgn). This includes all free day shifts, vacations shifts on weekends, etc., but not e.g. vacation shifts on Mon-Fri, as a physician is entitled to a certain number of rest periods every month in addition to vacation days.

G9: Working 1 if the shift is a working shift, otherwise 0.

G10: Start The earliest possible start time of a shift. There can be small variations with respect to the start time of shift depending on which day it is assigned. This variation is not relevant for this scheduling problem, and the start time is only used to ensure compliance with the Danish Working Time Directive.

G11: End The latest possible finish time of a shift. The explanation from 'G10: Start' also apply here.

G12: Hours The amount of working hours for the shift.

Week day compatibility 1 on the days in the week on which a shift is allowed to occur, otherwise zero.

Add-on compatibility 1 if a shift is allowed to be given in combination with a certain add-on. E.g. if there is a 1 in the cell connecting day shift 'D' with on call add-on shift 'AKC', then they are allowed to be assigned in combination with each other.

1.3.3 Data sheet 3: Shift Competencies

This sheet lists which competencies are required to man every shift and every add-on shift listed in *Data sheet 1: Personnel.* The list of possible competencies, Comp_1 to Comp_{25}, is the same list as in *Data sheet 1: Personnel.*

	Name:	NR	Comp_1	Comp_2	...	Comp_{25}
Add-ons	None		1			
	...			1	1	
	AKC					1
Shifts	D		1			
	...			1		
	FE_w		1			

Table 1.4: Simplified overview of the data in data sheet 3: Shift Competencies.

1.3.4 Data sheet 4: Wishes and shifts

This sheet maps which shift and add-on combinations are allowed based on a given wish. The physicians can only wish for a few specific shifts besides a limited set of broader wishes such as DAG - any day shift, VAGT - any night shift, etc. Here, the list of wishes used are the same as the one currently used by the anesthesia department. The only difference is that the wishes are now split into a shift wish and an add-on wish, e.g. the wish DL (long day shift) is equivalent to the wish combination of shift wish D and add-on wish L. In the wish sheet currently used by the anesthesia department, the physicians need to add a separate comment to a shift wish, if they e.g. are not able to man an on-call or alert shift after their shift wish. Unique comments are difficult to interpret for a computer and by splitting a wish into a shift wish and an add-on wish, some reoccurring comments can be avoided.

For any given wish, any add-on and shift combination for which both are marked with a '1', is allowed. Thus, considering the wish FRI (off day) in Table 1.5, there are $6 \cdot 2 = 12$ possible shift-add-on-combinations which fulfills the wish FRI. Even though the off shifts FRI_w and FRI_h are only allowed to be combined with the add-on 'None', this information is already encoded in *Data sheet 2: Shift types.*

For the wish -VAGT, no night/evening shift, combining shift DAG with any other add-on than 'None', results in a night/evening assignment. Therefore, no other add-on than 'None' must be allowed for the -VAGT wish. In general the logic that applies to shift wishes also applies to add-on wishes.

When wishing for both an add-on and a shift, then any add-on and shift combination which is allowed for both the shift wish and the add-on wish, is consequently allowed for the combination wish. Thus, for the combination wish: DAG and L, only the add-ons L_0 and L_475 are allowed in combination with the shift D, cf. the small example in Table 1.5.

	Wish:	No work	Work	Add-ons						Shifts					
				None	R_OLG	R_AFD	L_475	L_0	AKC	D	B2	FRI_h	FRI_w	...	ALN
Shift wish															
	FRI	1		1	1	1	1	1	1			1	1		
	...														
	DAG		1	1	1	1	1	1	1	1					
	-VAGT			1						1		1	1		
Add-on wish	-Alle			1						1	1	1	1	1	1
	R		1		1	1				1	1	1	1	1	1
	-R			1			1	1	1	1	1	1	1	1	1
	L		1				1	1		1	1	1	1	1	1
	-L			1	1	1			1	1	1	1	1	1	1
	AKC		1						1	1	1	1	1	1	1
	-AKC			1	1	1	1	1		1	1	1	1	1	1

Table 1.5: Simplified overview of the data in data sheet 4: Wishes and Shifts.

No work 1 if a shift wish is equivalent to wishing to be off, i.e. not working. This option is blocked out for add-ons as an add-on wish alone cannot specify that a physician wishes not to work.

Work 1 if a shift or add-on wish is equivalent to wishing to work, e.g. wishing for a day shift, 'D', an on-call add-on such as 'R', etc. Some wishes are neither marked as 'Not work' nor 'Work' such as -VAGT, i.e. not wanting a night/evening shift, because this wish implies neither.

1.3.5 Data sheet 5: Contracts

This sheet describes any special contracts of employment that each physician might have.

Counters \ Physician	Fixed num shifts			% of employment					Remainder %
	t_F1	t_ADM	t_TR	t_DIG	t_TC	t_CSV	...	t_LSR	
JDI	1	2							100
RAW				50					50
HJØ			2						100
CSV					70	30			0
LSR								100	0
SEA									100

Table 1.6: Simplified overview of the data in data sheet 5: Contracts.

Fixed num shifts consists of a set of counters, t_F1, t_ADM, and t_TR for which a physician can be entitled to a fixed number of shifts every month. E.g. physician JDI is contracted to receive at least two administrative days every month which are included in the counter t_ADM.

% of employment consists of a set of counters, t_DIG, t_TC, t_LSR, etc. for which a physician can be entitled to a fixed percentage of their employment every month. E.g. physician RAW is hired on a contract in which 50% of his employment is assigned to the digitalization unit, t_DIG. Every month, RAW's total employment hours are calculated based on the number of working days in the month, then the hours equivalent to any vacation, compensations days, etc. are deducted from his total employment hours as well as the hours equivalent to any 'Fixed num shift' contracts RAW might also have. Finally, 50% of the remaining hours will be assigned to the digitalization unit.

% Remainder is defined as; 100% $-\sum$(Fixed num shift) $-\sum$ (% of employment) $-\sum$(vacation wishes). This time can be assigned to any shift for which a physician has the competencies for. These shift primarily includes shift related the anesthesia clinic.

1.3.6 Data sheet 6: Counters

This sheet defines all the counters used to specify regular demands and demands on contracts. Furthermore, if fairness is applied to a counters, this is also specified in this sheet. A counter can count a certain competence, a shift/shifts, an add-on/add-ons or any combination of the above. The counter t_SPLG counts the number of specialists (chief physician or senior registrar) working in the clinic on any given day. Hence, the counter will count a physician who has the competency SPLG and is assigned a shift which is equivalent to working in the clinic, as seen in Table 1.7. There are a few assumptions to note about this data structure:

- For every counter there must be at least one competency, one add-on, and one shift allowed, i.e. marked by 1. This will always be possible, e.g. take the counter t_ADM (see Table 1.7). Here, we wish to count all instances of the shift ADM. An ADM shift requires no competencies, so the competence 'None' is selected as well as the shift ADM. It is irrelevant to this counter which add-on is assigned with the ADM shift, thus, shift-add-on-combinations where the shift is ADM are counted.
- Unlike shifts and add-ons, it is only possible to choose exactly one competency for each counter. This is because every physician is assigned exactly one shift and one add-on every day. The constraints which control the counters, count every instance of a counter, for which a physician is assigned both a valid shift and add-on and possesses the required competency. Thus, if two competencies are required, the same shift-add-on-combination will be counted twice. By creating new combinatorial competencies, it is still possible for a counter to require more than one competency or to require at least one competency from a set of competencies. E.g. the competency SPLG is assigned to a physician who is either chief physician or senior registrar.

Fair 1 if fairness is applied to the counter, otherwise zero. Fairness will attempt to assign an equal number of shifts-add-on-combination within the counter to every physician who has the competencies to take on at least one shifts-add-on-combination within the counter.

Counters\Comp, Add-ons & shifts		Fair	P / M	Competencies: None	Competencies: SPLG	Competencies: BØRN	Competencies: ...	Competencies: R	Add-ons: None	Add-ons: R_OLG	Add-ons: R_AFD	Add-ons: L_475	Add-ons: L_0	Add-ons: AKC	Shifts: D	Shifts: FRI_h	Shifts: ADM	Shifts: ...	Shifts: DIG
Fixed & Flex demand counters	t_SPLG		M		1				1	1	1	1	1	1	1			1	
	t_BØRN		M			1			1	1	1	1	1	1	1			1	
	t_R	1	P					1		1	1				1		1	1	1
	...		M				1		1	1	1	1	1	1				1	
	t_LANG	1	M				1					1	1		1			1	1
Fixed num shifts	t_F1		M	1					1	1	1	1	1	1				1	
	t_ADM		M	1					1	1	1	1	1	1			1		
	t_TR		M	1					1	1	1	1	1	1				1	
% of employment	t_DIG		M	1					1	1	1	1	1	1					1
	t_TC		M	1					1	1	1	1	1	1				1	
	...		M				1		1	1	1	1	1	1				1	
	t_LSR		M				1		1									1	

Table 1.7: Simplified overview of the data in data sheet 6: Counters

---

P / M is either P if a counter is a precise counter and M if the counter is a minimum counter:

- An example of a precise counter is t_R, the counter for the on-call shift 'R'. There should be exactly one physician manning this shift every day, not more or less.
- An example of a minimum counter t_SPLG, the counter for how many specialists are at work in the clinic. Depending on the amount of active hospital beds on a given day, there is a minimum number of specialists who must be at work.

1.3.7 Data sheet 7: Priorities

This sheet specifies if there are any shifts, add-ons, or competencies which should be prioritized within a counter.

Counter	Priority	1	2	3	4	...	10
t_8	Name	P8	TC8	D8			
	S / C / A	S					
t_RLG	Name	RLG					
	S / C / A	C					
...							
t_FRI	Name	FRI_w	FRI_h				
	S / C / A	S					

Table 1.8: Simplified overview of the data in data sheet 7: Priorities

Name is the list of names of the shifts, add-ons, or competencies which must be prioritized, listed in their prioritized order. E.g. the alert shift '8000', counted by the counter t_8, should preferably be covered by a P8 shift, then a TC8 shift, then a D8 shift, and otherwise any other shift within t_8.

S / C / A is S if the above listed priorities are shifts, C if they are competencies, and A if they are add-ons.

1.3.8 Data sheet 8: Fixed Demand

This sheet is for the scheduler to add reoccurring/fixed demand to the counters defined in Table 1.7, *Data sheet 6: Counters*. For the anesthesia department $\sim$ 23 counters are needed to define the standard demand setup used all year except during holidays. The fixed demand is specified based on even and uneven week numbers such that it does not need to be updated from month to month.
The demand is specified as the number of shifts/add-ons needed every day for each counter, where a blank field is the same as zero. Here, the precise / minimum counter definitions from Table 1.7 in *Data sheet 6: Counters* are important. As an example the minimum counter t_BØRN, the number of pediatricians, is zero on weekends (see Table 1.9). This does not exclude physicians with the BØRN competency to work on weekends. On the contrary, for the precise counter t_B1, a weekend day shift in the trauma center, there must be exactly zero B1 shifts every day of the week except on Saturday and Sunday where there must be exactly one.

		Even week number							Uneven week number						
Counters		Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Fixed & Flex demand counters	t_R	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	t_BØRN	3	3	3	3	3			3	3	3	3	3		
	t_D18	1							1						
	...														
	t_B1						1	1						1	1
Fixed num shifts	t_F1														
	t_ADM														
	t_TR														
% of employment	t_DIG			1	1	1			1	1					
	...														
	t_LSR														

Table 1.9: Simplified overview of the data in data sheet 8: Fixed Demand

1.3.9 Data sheet 9: Flex Demand

This sheet is for the scheduler to add flexible demand to the counters defined in Table 1.7, *Data sheet 6: Counters*. Currently, the anesthesia department only has one type of demand that varies from month to month. This is the minimum number of specialists at work in the clinic, counted by t_SPLG. The minimum number of specialists needed on each day is based on the number of scheduled active bed rests. The number of bed rests is planned across multiple departments at Rigshospitalet and is among others based on planned surgeries, holidays, etc.

Week number		26	26	26	26	27	27	27	...	30	30	30	30	30	30	30
Date		1	2	3	4	5	6	7	...	25	26	27	28	29	30	31
Holiday (H)									...							
Counters		Thu	Fri	Sat	Sun	Mon	Tue	Wed	...	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Fixed & Flex demand counters	t_SPLG	12	11			12	12	12			9	9	9	9	8	
	t_BØRN															
	t_D18															
	...															
	t_B1															
Fixed num shifts	t_F1															
	t_ADM															
	t_TR															
% of employment	t_DIG															
	...															
	t_LSR															

Table 1.10: Simplified overview of the data in data sheet 9: Flex Demand

1.3.10 Data sheet 10: Total Demand

This sheet is of the exact same format as Table 1.10 in *Data sheet 9: Flex Demand*. The only difference is that in this sheet the flexible demand from 1.10 in section *Data sheet 8: Fixed Demand* is added to the fixed demand creating a total demand overview of the entire month for the the scheduler.

1.3.11 Data sheet 11: Rules

This data sheet defines rules as required by the unions FAS and YL. Chief physicians belong to the FAS union and senior registrars as well as physician trainees belong to the YL union. It is possible to make a local special agreements with each individual physician, which is why each physician has an individual set of rules in this data sheet.

Physician	R1	R2	R3	R4	R5	R6	R7	R8
JDI	11	6	32	55	C	6	3	3
...	11	6	32	55	C	6	3	3
RAW	11	6	32	55	C	6	3	3

Table 1.11: Simplified overview of the data in data sheet 11: Rules

R1 is the minimum number of hours required from the end of a shift until the beginning a new shift.

R2 is the maximum number of work days in a row a physician can have before she or he must have at least a short rest period (see definition of a short rest period in R3 below).

R3 is the minimum number of consecutive free hours required for a short rest period.

R4 is the minimum number of consecutive free hours required for a normal rest period.

R5 is either C, P, or blank. After an alert or on-call shift during the night, a physician is entitled to a sleep shift 'SOV'. However, as described in section *Data sheet 2: Shift types*, G5, administrative shifts can follow an alert or on-call shift if a local agreement has been made. If R5 is blank, no agreement has been made, if R5 is 'C' then a physician *can* take on an administrative shift the following day but does not prefer it, and if R5 is 'P' then a physician *prefers* to have an administrative shift following an alert or on-call shift.

R6 During a norm period (3-12 months) a physician must at most have one presence shift every R6 (six) days on average.

R7 During a norm period (3-12 months) a physician must at most have one on-call shift every R7 (three) days on average.

R8 During a norm period (3-12 months) a physician must at most have one alert shift every R8 (three) days on average.

OLG max is 2/3. Defined as the maximum fraction of all presence shifts in a month that can be manned by chief specialists (OLG).

1.3.12 Data sheet 12: History and Targets

This sheet gives an overview of certain targets the scheduler can adjust for each physician. Furthermore, for the convenience of the scheduler, certain shifts, e.g. compensation days, which a physician is entitled to have a set amount of over the course of a longer time period, are counted.

Work days in month: 22	Targets				History		
Holidays in month: 0 Resultant norm hours : 162.8							
Physician	% of norm	Hours	Free periods	Hours status	KO	PKL	OM
BDY	80	130.2	3	2	0	0	0
SLN	100	162.8	3	2	5	0	0
...							
SWA	100	162.8	3	1	0	0	0

Table 1.12: Simplified overview of the data in data sheet 12: History and Targets

Work days in month is the number of days in the month excluding weekends.

Holidays in month is the number of national holidays (Mon-Fri only) in the month.

Resultant norm hours is: (Work days in month − Holidays in month)·7.4 hours, which is the amount of hours a physician should work if on a regular contract of 100% employment (see % of norm).

% of norm is the employment percentage of a physician with respect to full time.

Hours is the target number of hours a physician should hit this month.

Free periods is the targeted minimum number of free periods this month for a physician. Here, a short free period counts as 0.5 and a regular free period counts a 1.

Hours status 1, 2, or 3. If a physician has close to no over hours in the current norm period, then his/her status should be 1, if a physician has moderately too many over hours, then his/her status should be 2, and if a physician has far too many over hours, then his/her status should be 3. Here, the model will first try to assign necessary over hours to physicians with status 1, then 2, and lastly status 3.

History is the number of KO, PKL, and OM shifts given in the current time frame (year) to a physician. E.g. physicians with children under the age of 8 are entitled to two 'OM' shifts, a child care day, per year.

1.3.13 Data sheet 13: Wishes

This sheet shows the wishes of each physician on every day of the month. A physician can make three types of wishes every day: a shift wish, an add-on wish, and a priority wish. It is possible to wish for any combination of the three wish types.

Week num	26			26			...	30		
	Shift	Add-on	Priority	Shift	Add-on	Priority	...	Shift	Add-on	Priority
Date	1			2			...	31		
Week day	Thursday			Wednesday			...	Saturday		
Holiday							...			
JDI				DAG	-Alle				FE	*
JST	B			DAG	L			-VAGT		
...										
MEG	DAG	-L		DAG	-L	*				

Table 1.13: Simplified overview of the data in data sheet 13: Wishes

Shift wish is a wish for a specific shift or shift group. There are a list of 34 wishes to choose from in the anesthesia department. This list of possible wishes is the same list as the anesthesia department used prior to this thesis.

Add-on wish is a wish for a specific add-on or add-on group. There are a list of 7 wishes to choose from.

Priority wish is a wish priority and is either blank or *. A blank priority means that the wish has no priority and a * priority means that the wish will always be fulfilled. Star priorities are primarily used for holiday wishes, important work shifts, etc. It would be possible to add more priority options such as high and low, a priority points system, or similar.

2 Base model

The base model fulfills the various demands in the anesthesia department at Rigshospitalet. This means the model satisfy the demand on; specific shifts, groups of shifts, and personnel qualifications such as the number of present pediatrician.
The model ensures that the various demands are met while aiming to assign work equivalent to the target hours of every physician's contract (see *Data sheet 12: History and Targets*). Furthermore, every physician is assigned exactly one shift every day, including free shifts such as 'off' and 'vacation', as well as an add-on shift which includes a 'no' add-on option. A shift-add-on-combination is only assigned to a qualified physicians on a day in the week on which both the shift and the add-on are allowed to occur.
Lastly, the model tries to assign shifts in accordance with the contracted division of hours for each physician. As an example, physician RAW is contracted to work 50% of his time in the digitization department and the remaining 50% in the anesthesia clinic. Thus, the model aims to assign half of RAW's target hours to digitalization shifts and the other half to shifts that lie within the anesthesia clinic.

Sets:

$\mathcal{P}$	Set of all physicians.
$\mathcal{D}$	Set of all days in the roster.
$\mathcal{S}$	Set of all regular shifts including 'off' and vacation shifts.
$\mathcal{A}$	Set of all add-on shifts including a 'no add-on' option.
$\mathcal{C}$	Set of all counters.
$\mathcal{C}^{\text{perc}}$	Set of contracted counters, $\mathcal{C}^{\text{perc}} \subseteq \mathcal{C}$, which a physician can be assigned a percentage of their working hours to.
$\mathcal{C}^{\text{days}}$	Set of contracted counters, $\mathcal{C}^{\text{days}} \subseteq \mathcal{C}$, which a physician can be assigned a specific number of shifts to.
$\mathcal{C}^{\text{prio}}$	Set of counters, $\mathcal{C}^{\text{prio}} \subseteq \mathcal{C}$, for which certain shifts, competencies, or add-ons should be assigned in a prioritized order.
$\mathcal{V}$	Set of days in a week; Mon-Sun.
$\mathcal{K}$	Set of competencies a physicians can have.
$\mathcal{W}^{\text{add}}$	Set of all possible add-on wishes.
$\mathcal{W}^{\text{shi}}$	Set of all possible shift wishes.
$\mathcal{W}^{\text{prio}}$	Set of all possible wish priorities.

Languages:

$\mathcal{L}$	Set of all possible combinations of letters in the Danish alphabet, i.e. a word.

Parameters:

NR_p	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ is a normal resource in the clinic, otherwise 0.
St_p^{emp}	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ is inactive in the clinic, e.g. on maternity leave, otherwise 0.
St_s^{shi}	$\mathbb{Z}_2$	1 if regular shift $s \in \mathcal{S}$ is inactive, i.e. not used, otherwise 0.
H_s^{shi}	$\mathbb{R}^{0+}$	Number of work hours for each shift $s \in \mathcal{S}$.
H_a^{add}	$\mathbb{R}^{0+}$	Number of work hours for each add-on $a \in \mathcal{A}$.
H_p^{targ}	$\mathbb{R}^{0+}$	Number of total target work hours for physician $p \in \mathcal{P}$.
$D_{s,v}^{\text{shi}}$	$\mathbb{Z}_2$	1 if shift $s \in \mathcal{S}$ can occur on week day $v \in \mathcal{V}$, otherwise 0.
D_d^{num}	$\mathbb{Z}_7^+$	1 if day $d \in \mathcal{D}$ is a Monday, 2 if $d \in \mathcal{D}$ is a Tuesday, etc.
C_c^{type}	$\mathcal{L}$	"P" if $c \in \mathcal{C}$ is an equality/precise counter, "M" if c is a minimum counter.
$K_{c,k}^{\text{cou}}$	$\mathbb{Z}_2$	1 if counter $c \in \mathcal{C}$ requires competency $k \in \mathcal{K}$, otherwise 0.
$K_{p,k}^{\text{emp}}$	$\mathbb{Z}_2$	1 if physician $p \in$ has competency $k \in \mathcal{K}$, otherwise 0.
$K_{s,k}^{\text{shi}}$	$\mathbb{Z}_2$	1 if shift $s \in \mathcal{S}$ requires competency $k \in \mathcal{K}$, otherwise 0.
$K_{a,k}^{\text{add}}$	$\mathbb{Z}_2$	1 if add-on $a \in \mathcal{A}$ requires competency $k \in \mathcal{K}$, otherwise 0.
$SA_{s,a}^{\text{com}}$	$\mathbb{Z}_2$	1 if shift $s \in \mathcal{S}$ is compatible with add-on $a \in \mathcal{A}$, otherwise 0.
$C_{c,a}^{\text{add}}$	$\mathbb{Z}_2$	1 if counter $c \in \mathcal{C}$ permits add-on $a \in \mathcal{A}$, otherwise 0.
$C_{c,s}^{\text{shi}}$	$\mathbb{Z}_2$	1 if counter $c \in \mathcal{C}$ permits regular shift $s \in \mathcal{S}$, otherwise 0.
$C_{c,d}^{\text{dem}}$	$\mathbb{Z}^{0+}$	The demand for counter $c \in \mathcal{C}$ of day $d \in \mathcal{D}$.
$C_{p,c}^{\text{days}}$	$\mathbb{Z}^{0+}$	The number of monthly contracted days physician $p \in \mathcal{P}$ must have of counter $c \in \mathcal{C}^{days}$.
$C_{p,c}^{\text{perc}}$	$\mathbb{R}^{0+}$	The percentage of the target monthly hours for a physician $p \in \mathcal{P}$ which should be assigned to counter $c \in \mathcal{C}^{\text{perc}}$.
C_c^{prio}	$\mathcal{L}$	"Shift", "Add-on", or "Competency" depending on whether prioritized counter $c \in \mathcal{C}^{\text{prio}}$ contains a list of either prioritized shifts, add-ons, or competencies.
$PS_{p,s}^{\text{pos}}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ has all the competencies a shift $s \in \mathcal{S}$ requires, otherwise 0.
$PA_{p,s}^{\text{pos}}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ has all the competencies an add-on $a \in \mathcal{A}$ requires, otherwise 0.
$C_{p,c,s,a}^{\text{pos}}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ has; the single competency $k \in \mathcal{K}$ required for counter $c \in \mathcal{C}$, the competencies required for both shift $s \in \mathcal{S}$ and add-on $a \in \mathcal{A}$, if shift s and add-on a both are valid for counter c, and if shift s and add-on a are comptaible with each other, otherwise 0. This parameter is calculated as $C_{c,p,s,a}^{\text{pos}} = \sum_{k \in \mathcal{K}} \Big(K_{c,k}^{\text{cou}} \; K_{p,k}^{\text{emp}} \; C_{c,a}^{\text{add}} \; C_{c,s}^{\text{shi}} \; PS_{p,s}^{\text{pos}} \; PA_{p,a}^{\text{pos}} \; SA_{s,a}^{\text{com}} \Big)$.

S^{oth}_{s}	$\mathbb{Z}_2$	1 if a shift $s \in \mathcal{S}$ neither counts towards a contract nor the clinic, e.g. a vacation shift. Equivalent to G6 in *Data sheet 2: Shift types.*
$WS^{\text{add}}_{w,a}$	$\mathbb{Z}_2$	1 if add-on $a \in \mathcal{A}$ is allowed given shift wish $w \in \mathcal{W}^{\text{shi}}$, otherwise 0.
$WS^{\text{shi}}_{w,s}$	$\mathbb{Z}_2$	1 if shift $s \in \mathcal{S}$ is allowed given shift wish $w \in \mathcal{W}^{\text{shi}}$, otherwise 0.
$WA^{\text{add}}_{w,a}$	$\mathbb{Z}_2$	1 if add-on $a \in \mathcal{A}$ is allowed given add-on wish $w \in \mathcal{W}^{\text{add}}$, otherwise 0.
$WA^{\text{shi}}_{w,s}$	$\mathbb{Z}_2$	1 if shift $s \in \mathcal{S}$ is allowed given add-on wish $w \in \mathcal{W}^{\text{add}}$, otherwise 0.
$WP^{\text{shi}}_{p,d}$	$\mathcal{L}$	The shift wish of physician $p \in \mathcal{P}$ on day $d \in \mathcal{D}$.
$WP^{\text{add}}_{p,d}$	$\mathcal{L}$	The add-on wish of physician $p \in \mathcal{P}$ on day $d \in \mathcal{D}$.
$WP^{\text{prio}}_{p,d}$	$\mathcal{L}$	The priority wish of physician $p \in \mathcal{P}$ on day $d \in \mathcal{D}$.
$PP^{\text{shi}}_{c,s}$	$\mathbb{R}^{0+}$	The punish value for each shift $s \in \mathcal{S}$ for every counter $c \in \mathcal{C}^{\text{prio}}$.
$PP^{\text{add}}_{c,a}$	$\mathbb{R}^{0+}$	The punish value for each add-on $a \in \mathcal{A}$ for every counter $c \in \mathcal{C}^{\text{prio}}$.
$BP^{\text{comp}}_{c,k}$	$\mathbb{R}^{0+}$	The benefit value for each competence $k \in \mathcal{K}$ for every counter $c \in \mathcal{C}^{\text{prio}}$.
St^{oHr}_{p}	$\mathbb{Z}^{+}_{3}$	The over hours status for physician $p \in \mathcal{P}$. See 'Hours status' in *Data sheet 12: History and Targets.*
$H^{\text{frac}}_{p,s}$	$\mathbb{R}_{[0,1]}$	A fraction equivalent to the percentage of norm employment of each physician $p \in \mathcal{P}$ (see Table 1.12) if and only if *G7: Contract dependency* (see Table 1.3) is 1 for shift $s \in \mathcal{S}$, otherwise 1.

Decision variables:

$x_{p,d,s,a}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ is assigned regular shift $s \in \mathcal{S}$ in combination with add-on shift $a \in \mathcal{A}$ on day $d \in \mathcal{D}$, otherwise 0.
$x^{\text{help}}_{p,s,a}$	$\mathbb{Z}^{0+}$	The number of regular shift $s \in \mathcal{S}$ in combination with add-on shift $a \in \mathcal{A}$, that physician $p \in \mathcal{P}$ is assigned this month.
h^{ov}_{p}	$\mathbb{R}^{0+}$	The number of hours over target H^{targ}_{p} for physician $p \in \mathcal{P}$.
h^{un}_{p}	$\mathbb{R}^{0+}$	The number of hours under target H^{targ}_{p} for physician $p \in \mathcal{P}$.
$o^{\text{dem}}_{c,d}$	$\mathbb{Z}^{0+}$	The over demand on counter $c \in \mathcal{C}$ on day $d \in \mathcal{D}$.
$u^{\text{dem}}_{c,d}$	$\mathbb{Z}^{0+}$	The under demand on counter $c \in \mathcal{C}$ on day $d \in \mathcal{D}$.
$hC^{\text{ov}}_{p,c}$	$\mathbb{R}^{0+}$	The number of hours over contracted percentage target $C^{\text{perc}}_{p,c}$ for physician $p \in \mathcal{P}$ and counter $c \in \mathcal{C}^{\text{perc}}$.
$hC^{\text{un}}_{p,c}$	$\mathbb{R}^{0+}$	The number of hours under contracted percentage target $C^{\text{perc}}_{p,c}$ for physician $p \in \mathcal{P}$ and counter $c \in \mathcal{C}^{\text{perc}}$.
p^{prio}_{c}	$\mathbb{R}^{0+}$	The sum of all punishments associated with counter $c \in \mathcal{C}^{\text{prio}}$.

2.1 Objective

The objective minimizes the total over and under hours for all physicians with respect to both their target working hours, H_p^{targ}, and their contracted working hours, H_p^{perc}. Here, the over hour status St_p^{oHr} is multiplied with the decision variable for over hours, h_p^{ov}, such that it is more favorable to give over hours to physicians with a low status. The over and under hours constraints are defined in *General model constraints*. Secondly, the objective minimizes the punishment given when not respecting a prioritized counter. The related constraints are defined in *Employment contracts*. Thirdly, the objective minimizes the punishment, p_c^{prio}, associated with counters with a priority list including either shifts, add-ons, or competencies. The related constraints are defined in *Priority counters*. Lastly, to avoid infeasibility, a slack variable for under and over demand, $o_{c,d}^{\text{dem}}$ and $u_{c,d}^{\text{dem}}$, respectively, with a high punishment is also added to the objective. The demand constraints are defined in *Demands*.

$$\begin{aligned} \text{Min} \quad & \sum_{p\in\mathcal{P}} (St^{\text{oHr}} h_p^{\text{ov}} + 10 h_p^{\text{un}}) + \sum_{p\in\mathcal{P}} \sum_{c\in\mathcal{C}^{\text{perc}}} (hC_{p,c}^{\text{ov}} + 5hC_{p,c}^{\text{un}}) \\ & + \sum_{c\in\mathcal{C}^{\text{prio}}} p_c^{\text{prio}} + 500 \sum_{c\in\mathcal{C}} \sum_{d\in\mathcal{D}} \left(o_{c,d}^{\text{dem}} + u_{c,d}^{\text{dem}} \right) \end{aligned} \tag{O1}$$

2.2 General model constraints

A physician p which has the status inactive, $St_p^{\text{emp}} = 1$, must not be assigned any work:

$$\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} St_p^{\text{emp}} \, x_{p,d,s,a} = 0, \qquad \forall p \in \mathcal{P}, \forall d \in \mathcal{D} \tag{C1.1}$$

Every physician p should be assigned exactly one shift-add-on-combination every day if the physician is active, $St_p^{\text{emp}} = 0$:

$$\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} x_{p,d,s,a} = 1, \qquad \forall p \in \mathcal{P}, \forall d \in \mathcal{D};\ St_p^{\text{emp}} = 0 \tag{C1.2}$$

A shift cannot be assigned if the shift is inactive, $St_p^{\text{shi}} = 1$, hence not currently in use:

$$\sum_{a\in\mathcal{A}} St_p^{\text{shi}} \, x_{p,d,s,a} = 0, \qquad \forall p \in \mathcal{P}, \forall d \in \mathcal{D}, \forall s \in \mathcal{S} \tag{C1.3}$$

Determine the total over-time for the entire schedule for each physician p based on all shift and add-on shift assignments. The overtime, h_p^{ov}, is calculated as the sum of all shifts and add-on hours minus the monthly target, H_p^{targ}. Here, $H_{p,s}^{\text{frac}}$ ensures that physicians employed $< 100\%$ of norm target hours, receive equivalently less hours for shifts such as a vacation day:

$$\sum_{d\in\mathcal{D}} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} x_{p,d,s,a} \, (H_{p,s}^{\text{frac}} \ H_s^{\text{shi}} + H_a^{\text{add}}) - H_p^{\text{targ}} \leq h_p^{\text{ov}}, \qquad \forall p \in \mathcal{P} \tag{C1.4}$$

Equivalently, the total under-time for the entire schedule for each physician p can be determined as:

$$H_p^{\text{targ}} - \sum_{d\in\mathcal{D}} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} x_{p,d,s,a} (H_{p,s}^{\text{frac}} \ H_s^{\text{shi}} + H_a^{\text{add}}) \leq h_p^{\text{un}}, \qquad \forall p \in \mathcal{P} \tag{C1.5}$$

Ensure a shift s can only assigned to a week day, v, where s is allowed to occur. Here, $v = D_d^{\text{num}} \in \{1...7\}$ stores the week day number for each date d in the schedule, such that $D_{s,v}^{\text{shi}}$ is 1 if shift s can occur on week day v:

$$\sum_{a\in\mathcal{A}} x_{p,d,s,a} \leq D_{s,v}^{\text{shi}}, \qquad \forall p \in \mathcal{P},\, \forall d \in \mathcal{D},\, \forall s \in \mathcal{S},\; v = \{D_d^{\text{num}}\} \tag{C1.6}$$

A physician p can only be a assigned a shift s, she or he has all the competencies k for:

$$K_{p,k}^{\text{emp}} \geq x_{p,d,s,a}\; K_{s,k}^{\text{shi}}, \qquad \forall p \in \mathcal{P},\, \forall d \in \mathcal{D},\, \forall s \in \mathcal{S},\, \forall a \in \mathcal{A},\, \forall k \in \mathcal{K} \tag{C1.7}$$

A physician p can only be a assigned an add-on shift a, she or he has all the competencies k for:

$$K_{p,k}^{\text{emp}} \geq x_{p,d,s,a}\; K_{a,k}^{\text{add}}, \qquad \forall p \in \mathcal{P},\, \forall d \in \mathcal{D},\, \forall s \in \mathcal{S},\, \forall a \in \mathcal{A},\, \forall k \in \mathcal{K} \tag{C1.8}$$

An add-on a can only be assigned in combination with a shift s which it is compatible with:

$$x_{p,d,s,a} \leq\; SA_{s,a}^{\text{com}}, \qquad \forall p \in \mathcal{P},\, \forall d \in \mathcal{D},\, \forall s \in \mathcal{S},\, \forall a \in \mathcal{A} \tag{C1.9}$$

2.3 Demands

Precise demands (represented by counters $c \in \mathcal{C}$ where $C_c^{\text{type}} = \{P\}$) must be met exactly every day, e.g. there must be exactly one person on stand-by every day. Here, demands can be specified with respect to both physician competencies, add-ons, and regular shifts and any combination of the three; e.g. a pediatrician (competency) who must have a long day shift, D+L, on a Thursday, where D is a regular day shift and L is a long shift. The above is ensured by multiplying together: $C_{p,c,s,a}^{\text{pos}}$, whether physician p has the competencies to be assigned at least one valid shift-add-on-combination for counter c, and the decision variable $x_{p,d,s,a}$ for whether or not the shift-add-on-combination is given, to whom and when. Here, $C_{p,c,s,a}^{\text{pos}}$ is defined as the multiplication of the single competency k required for counter c, the set of competencies which every physician p has, the set of allowed add-ons a within the scope of counter c, the set of allowed shifts s within the scope of counter c, whether shift s and add-on a are compatible, and lastly, if physician p has all the competencies required for both shift s and add-on a. By requiring exactly one competency k for each counter c as well as assigning exactly one shift s and one add-on a every day, the constraint will count every shift-add-on-combination within the counter c exactly once.
To avoid infeasibility, a slack variable for both under and over demand, $o_{c,d}^{\text{dem}}$ and $u_{c,d}^{\text{dem}}$, respectively, are also added:

$$\begin{gathered}\sum_{p\in\mathcal{P}} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} C_{p,c,s,a}^{\text{pos}}\; x_{p,d,s,a} = C_{c,d}^{\text{dem}} + o_{c,d}^{\text{dem}} - u_{c,d}^{\text{dem}} \\ \forall d \in \mathcal{D}, \forall c \in \mathcal{C},\, C_c^{\text{type}} = \{P\}\end{gathered} \tag{C1.10}$$

Using the same logic, it can similarly be ensured that minimum demands (represented with counters $c \in \mathcal{C}$ where $C_c^{\text{type}} = \{M\}$) are also met every day, e.g. there must be at least 16 specialists working in the clinic on a Monday.
Again, to avoid infeasibility, the slack variable for under demand, $u_{c,d}^{\text{dem}}$, is added:

$$\begin{gathered}\sum_{p\in\mathcal{P}} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} C_{p,c,s,a}^{\text{pos}}\; x_{p,d,s,a} \geq C_{c,d}^{\text{dem}} - u_{c,d}^{\text{dem}} \\ \forall d \in \mathcal{D}, \forall c \in \mathcal{C},\, C_c^{\text{type}} = \{M\}\end{gathered} \tag{C1.11}$$

2.4 Employment contracts

To simplify certain constraints, a helper variable $x^{\text{help}}_{p,s,a}$ is first defined. Here, $x^{\text{help}}_{p,s,a}$ is equal to the total number of shift-add-on combinations for physician p for the entire schedule, for all shifts s and add-ons a:

$$\sum_{d \in \mathcal{D}} x_{p,d,s,a} = x^{\text{help}}_{p,s,a}, \qquad \forall p \in \mathcal{P},\ \forall s \in \mathcal{S},\ \forall a \in \mathcal{A} \tag{C1.12}$$

Every physician p is assigned at least his or her contracted fixed number of shifts or add-ons, $C^{\text{days}}_{p,c}$, as stated in *Data sheet 5: Contracts*. E.g. physician p is contracted to at least three administration days per schedule:

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} C^{\text{shi}}_{c,s}\ C^{\text{add}}_{c,a}\ x^{\text{help}}_{p,s,a} \geq C^{\text{days}}_{p,c}, \qquad \forall p \in \mathcal{P}, \forall c \in \mathcal{C}^{\text{days}} \tag{C1.13}$$

Every physician p is assigned the percentage of their total monthly hours to the sub-departments or groups of shifts that are stated in their contracts in *Data sheet 5: Contracts*. As an example, a physician p is assigned 40% of his or her time to the trauma center and the remaining 60% to the anesthesia clinic. If physician p is also contracted a fixed number of shifts, as ensured by (C1.13), then the hours equivalent to these shifts are first deducted from the total target hours, H^{targ}_p. Furthermore, any shifts in the category 'other' (see G6 in *Data sheet 2: Shift types*), such as vacation (FE_h) which accounts for 7.4 hours, are also deducted. Then, the remaining hours are split into the e.g. 40%/60% split as illustrated in Figure 2.1. Here, both over as well as under hours with respect to the percentage contracts, $hC^{\text{ov}}_{p,c}$ and $hC^{\text{un}}_{p,c}$, are minimized in the objective.
Fixed contracted shifts, $C^{\text{days}}_{p,c}$, can be legally assigned to a group of shifts with varying duration, e.g. an administrative shift with a duration 7.4 hours and another with 7.5 hours. Therefore, the duration of all shifts which count towards a fixed shifts contract, are averaged before multiplied with the contracted frequency and then deducted from the total target hours, H^{targ}_p.

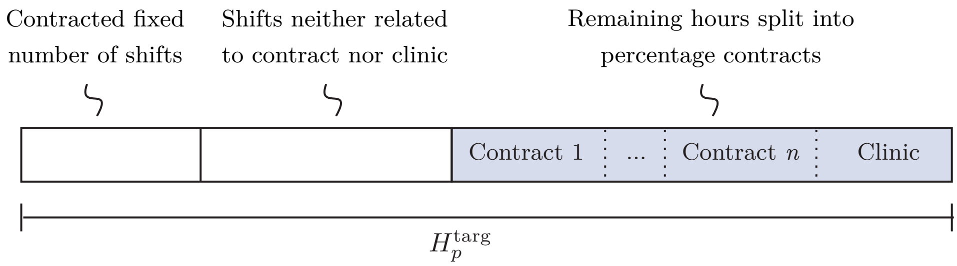

Figure 2.1: Illustration of which hours make up the basis for percentage contracts in (C1.14) and (C1.15) and how they are calculated. The size of the bars pictured are not representative of a typical time distribution for a physician.

The under hours with respect to % of employment contracts, are consequently minimized by:

$$\left(H^{\text{targ}}_p - \sum_{c_1 \in \mathcal{C}^{\text{days}}} C^{\text{dem}}_{c_1,d} \frac{\sum_{s_1 \in \mathcal{S}} C^{\text{shi}}_{c_1,s_1}\ H^{\text{frac}}_{p,s_1}\ H^{\text{shi}}_{s_1}}{\sum_{s_2 \in \mathcal{S}} C^{\text{shi}}_{c_1,s_2}} - \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} x^{\text{help}}_{p,s,a}\ S^{\text{oth}}_s\ H^{\text{frac}}_{p,s}\ H^{\text{shi}}_s \right) \frac{C^{\text{perc}}_{p,c}}{100}$$
$$\leq \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left(x_{p,d,s,a}\ C^{\text{shi}}_{c,s}\ H^{\text{frac}}_{p,s}\ H^{\text{shi}}_s \right) + hC^{\text{un}}_{p,c}, \qquad \forall p \in \mathcal{P}, \forall c \in \mathcal{C}^{\text{perc}}$$
$$\tag{C1.14}$$

The over hours with respect to percentage of employment contracts, are similarly minimized by:

$$\left(H_p^{\text{targ}} - \sum_{c_1 \in \mathcal{C}^{\text{days}}} C_{c_1,d}^{\text{dem}} \frac{\sum_{s_1 \in \mathcal{S}} C_{c_1,s_1}^{\text{shi}} H_{p,s_1}^{\text{frac}} H_{s_1}^{\text{shi}}}{\sum_{s_2 \in \mathcal{S}} C_{c_1,s_2}^{\text{shi}}} - \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} x_{p,s,a}^{\text{help}} S_s^{\text{oth}} H_{p,s}^{\text{frac}} H_s^{\text{shi}}\right) \frac{C_{p,c}^{\text{perc}}}{100}$$
$$\geq \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left(x_{p,d,s,a} C_{c,s}^{\text{shi}} H_{p,s}^{\text{frac}} H_s^{\text{shi}}\right) - hC_{p,c}^{\text{ov}}, \qquad \forall p \in \mathcal{P}, \forall c \in \mathcal{C}^{\text{perc}} \tag{C1.15}$$

2.5 Wishes

Wishes marked with a priority star, *, must be fulfilled, and are thus added as hard constraints. Here, we distinguishes between three star-wish scenarios: the wish only contains a shift wish, the wish only contains an add-on wish, and the wish contains both a shift and an add-on wish.

If the wish w of physician p has a star priority and the wish contains a shift wish only, then both the shift and add-on assigned, must be allowed given the shift wish, $WP_{p,d}$. Thus, $WS_{w,a}^{\text{add}}$ and $WS_{w,a}^{\text{shi}}$ must both be 1:

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} x_{p,d,s,a} WS_{w,a}^{\text{add}} WS_{w,s}^{\text{shi}} = 1,$$
$$\forall p \in \mathcal{P},\ \forall d \in \mathcal{D},\ w = \{WP_{p,d}^{\text{shi}}\};\ WP_{p,d}^{\text{prio}} = \{*\} \wedge WP_{p,d}^{\text{shi}} \neq \emptyset \wedge WP_{p,d}^{\text{add}} = \emptyset \tag{C1.16}$$

The same logic applies when a star-wish only contains an add-on wish:

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} x_{p,d,s,a} WA_{w,a}^{\text{add}} WA_{w,s}^{\text{shi}} = 1,$$
$$\forall p \in \mathcal{P},\ \forall d \in \mathcal{D},\ w = \{WP_{p,d}^{\text{add}}\};\ WP_{p,d}^{\text{prio}} = \{*\} \wedge WP_{p,d}^{\text{shi}} = \emptyset \wedge WP_{p,d}^{\text{add}} \neq \emptyset \tag{C1.17}$$

Lastly, when a star-wish contains both a shift and an add-on wish, then a shift-add-on-combination is allowed if $WS_{w,a}^{\text{add}}$, $WS_{w,a}^{\text{shi}}$, $WA_{w,a}^{\text{add}}$ and $WA_{w,a}^{\text{shi}}$ are all equal to 1 (see *Data sheet 4: Wishes and shifts*). Hence, given the shift wish, the assigned shift and add-on must both be allowed and given the add-on wish, the assigned shift and add-on must also both be allowed. As an example, for the shift wish 'DAG', a day shift in the clinic, all add-ons are allowed as well as the set of shift which are day shifts in the clinic. For the add-on wish 'R', a stand-by shift during the night, only the add-on 'R' is allowed as well as all other regular shift which are compatible with the add-on shift 'R'. Thus, when wishing for the combination 'DAG+R' then the only add-on allowed is 'R' and the only shifts allowed are the day shifts in the clinic which are also compatible with the 'R' add-on:

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} x_{p,d,s,a} WS_{w_1,a}^{\text{add}} WS_{w_1,s}^{\text{shi}} WA_{w_2,a}^{\text{add}} WA_{w_2,s}^{\text{shi}} = 1, \qquad \forall p \in \mathcal{P},\ \forall d \in \mathcal{D},$$
$$w_1 = \{WP_{p,d}^{\text{shi}}\},\ w_2 = \{WP_{p,d}^{\text{add}}\};\ WP_{p,d}^{\text{prio}} = \{*\} \wedge WP_{p,d}^{\text{shi}} \neq \emptyset \wedge WP_{p,d}^{\text{add}} \neq \emptyset \tag{C1.18}$$

2.6 Priority counters

The last set of constraints concerns the prioritization between certain shifts, add-ons, and competencies. As an example, the shift 'LL' is a basic function and whenever possible, the anesthesia department would like to assign a trainee to the 'LL' shift rather than a fully educated physician. Similarly, there exists a prioritized list of shifts for how to best cover

the stand-by function '8000'. This stand-by function can be covered during the day by one of the three shifts; P8, TC8, and D8. These shifts are listed in the order they should be prioritized. Priorities can also exist for add-ons, although there are currently none in the anesthesia department. This option has been implemented for scalability reasons.
The priorities are implemented as soft constraints. Here, the approach taken for a shift and add-on prioritization is to assign a punish value, $PP^{\text{shi}}_{c,a}$ and $PP^{\text{add}}_{c,a}$, between 0-1 to all shifts s and add-ons a, respectively, for every counter c which must be prioritized. The first priority is always assigned a zero, shifts/add-ons not in the priority list are assigned a one, and the punish value for the remaining shifts/add-ons in the priority list is equal to the priority number of the shift/add-on minus one, all divided by the total number of shifts/add-ons in the list. As an example, for the '8000' counter we assign the punish value $PP^{\text{shi}}_{c,s} = 0$ to the first priority shift, P8, the punish value $PP^{\text{shi}}_{c,s} = (2-1)/3 = 1/3$ to the second priority shift, TC8, the punish value $PP^{\text{shi}}_{c,s} = (3-1)/3 = 2/3$ to D8, and lastly, the punish value 1 to all other shifts.
Based on all shift and add-on assignments in the schedule, the total punish value, p^{prio}_c, for each prioritized counter c can be determined and consequently minimized in the objective (O1):

$$\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C^{\text{pos}}_{p,c,s,a}\; x^{\text{help}}_{p,s,a}\; PP^{\text{shi}}_{c,s} \leq p^{\text{prio}}_c, \qquad \forall c \in \mathcal{C}^{\text{prio}};\; C^{\text{prio}}_c = \{\text{Shift}\} \tag{C1.19}$$

$$\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C^{\text{pos}}_{p,c,s,a}\; x^{\text{help}}_{p,s,a}\; PP^{\text{add}}_{c,a} \leq p^{\text{prio}}_c, \qquad \forall c \in \mathcal{C}^{\text{prio}};\; C^{\text{prio}}_c = \{\text{Add-on}\} \tag{C1.20}$$

The same approach cannot be taken when prioritising on competencies. Here, every physician p has a unique set of competencies and by punishing the competencies that are not relevant to a certain prioritized counter, physicians with the most competencies will consequently be punished the most.
Instead, a benefit value, $BP^{\text{comp}}_{c,k}$, is defined for every competency k for each prioritized counter c. Similar to before, the assigned benefit value is between $0-1$. However, the benefit values are assigned in reverse compared to the punish values, such that the first priority competency receives the benefit value $BP^{\text{comp}}_{c,k} = 1$, all competencies not in the priority list receive the benefit value $BP^{\text{comp}}_{c,k} = 0$, and the benefit value of the remaining competencies in the priority list is equal to one minus; the priority number of the competency minus one, all divided by the total number of competencies in the list. Thus, the same calculation as for the punish values, now just subtracted from 1 to get the inverse.
Based on all competencies in the schedule, the total punish value, p^{prio}_c, for each prioritized counter c on competencies, can be determined and consequently minimized in the objective (O1). If a physician p possess the first priority competency with the benefit value 1, then consequently the punish value contribution from that physician will be zero in (C1.21). This way, the lower the sum of the benefit values of a physician is, the higher the punish value, p^{prio}_c, will be. Here, it is assumed that all competencies in any priority list are mutually exclusive, such that every physician possess at most one of these competencies:

$$\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C^{\text{pos}}_{p,c,s,a}\; x^{\text{help}}_{p,s,a} \left(1 - \sum_{k\in\mathcal{K}} K^{\text{emp}}_{p,k}\; BP^{\text{comp}}_{c,k}\right) \leq p^{\text{prio}}_c, \tag{C1.21}$$
$$\forall c \in \mathcal{C}^{\text{prio}};\; C^{\text{prio}}_c = \{\text{Competency}\}$$

3 MVP model

The minimum viable product (MVP) model is an extension of the *Base model*. This means that all constraints from the base-model, (C1.1)-(C1.21), are also part of the MVP model. The only exception is (C1.6). This constraint ensures that shifts and add-ons can only be assigned on a week day where the shift or add-on is allowed to occur. The MVP model now takes national holidays into account. The set of shifts and add-ons that can be assigned on a national holiday is the same set of shifts and add-ons that can be assigned on a day in the weekend. Here, instead of modifying the constraint to accommodate the exception, the data is modified instead. In data sheet from *Data sheet 10: Total Demand*, a given day in the schedule is marked by 'H' if it is a national holiday. Furthermore, every day is labelled with a number from 1-7, where one corresponds to Monday and seven is Sunday. If a day is a Monday-Friday and a national holiday, then the label of that day is simply changed to 7, i.e. a Sunday. This way (C1.6) will correctly assign only Sunday shifts on national holidays.

As the name of the MVP model suggests, the model fulfills all of the vital requirements that the scheduler at the anesthesia department has. Here, it has been chosen that only individual fairness for each physician is excluded from the MVP model. From a strategic standpoint, the MVP model is therefore sufficient as it meets all requirements. It is also financially the most optimal model as it maximizes the use of the available resources with respect to the lowest combined cost. However, employee satisfaction is not to be underestimated, as will be discussed once the full model, including fairness, has been introduced.

The additional sets, parameters, and decision variables needed to define the MVP model are presented below. Moreover, the definition of how we have chosen to refer to specific days in a roster is visually depicted in Figure 3.1. This definition is used throughout various constraints.

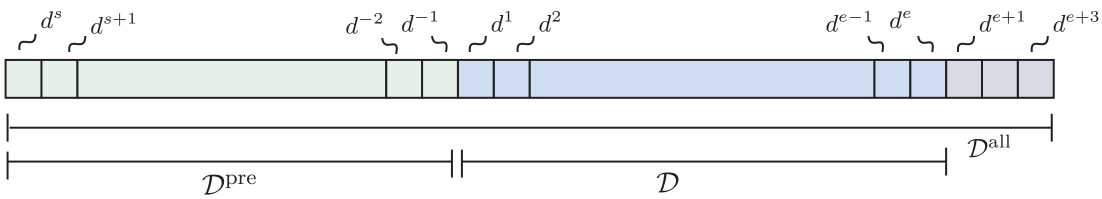

Figure 3.1: Illustration of how to refer to specific elements/days in the sets; $\mathcal{D}$, $\mathcal{D}^{\text{pre}}$, and $\mathcal{D}^{\text{all}}$.

Sets:

$\mathcal{D}^{\text{pre}}$ Set of all days in the previous month.

$\mathcal{D}^{\text{all}}$ Set of: all days in the previous month, all days in the current month (to be scheduled), and the first three days of the coming month.

$\mathcal{Q}$ Set of all weekends coinciding both fully and partially with the current month as well as the last non-coinciding weekend in the previous month.

Parameters:

$OP^{\text{shi}}_{p,d}$	$\mathcal{L}$	The name of the shift $s \in \mathcal{S}$ an employee $p \in \mathcal{P}$ was assigned each day $d \in \mathcal{D}^{\text{pre}}$ in the previous month.
$OP^{\text{add}}_{p,d}$	$\mathcal{L}$	The name of add-on $a \in \mathcal{A}$ an employee $p \in \mathcal{P}$ was assigned each day $d \in \mathcal{D}^{\text{pre}}$ in the previous month.
L^{shi}_{s}	$\mathcal{L}$	The list of the names of all shifts $s \in \mathcal{S}$.
$L^{\text{add-on}}_{s}$	$\mathcal{L}$	The list of the names of all add-ons $a \in \mathcal{A}$.
RA^{pres}_{a}	$\mathbb{Z}_2$	1 if add-on $a \in \mathcal{A}$ is tagged as a presence shift in 'G2: Rule type' in *Data sheet 2: Shift types*, otherwise 0.
RA^{alert}_{a}	$\mathbb{Z}_2$	1 if add-on $a \in \mathcal{A}$ is tagged as an alert shift in 'G2: Rule type' in *Data sheet 2: Shift types*, otherwise 0.
RA^{call}_{a}	$\mathbb{Z}_2$	1 if add-on $a \in \mathcal{A}$ is tagged as an on-call shift in 'G2: Rule type' in *Data sheet 2: Shift types*, otherwise 0.
RS^{pres}_{s}	$\mathbb{Z}_2$	1 if shift $s \in \mathcal{S}$ is tagged as a presence shift in 'G2: Rule type' in *Data sheet 2: Shift types*, otherwise 0.
RS^{alert}_{s}	$\mathbb{Z}_2$	1 if shift $s \in \mathcal{S}$ is tagged as an alert shift in 'G2: Rule type' in *Data sheet 2: Shift types*, otherwise 0.
RS^{call}_{s}	$\mathbb{Z}_2$	1 if shift $s \in \mathcal{S}$ is tagged as an on-call shift in 'G2: Rule type'in *Data sheet 2: Shift types*, otherwise 0.
R^{pref}_{p}	$\mathbb{Z}_2$	1 if a physician $p \in \mathcal{P}$ prefers to have an administrative shift instead of a sleep shift 'SOV' after an alert- or on-call-shift, otherwise 0. Equivalent to 'R5=P' in *Data sheet 11: Rules*.
R^{can}_{p}	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ can be assigned an administrative shift instead of a sleep shift SOV after an alert- or on-call-shift, otherwise 0. Equivalent to R5=C in *Simplified overview of the data in data sheet 11: Rules*.
S^{adm}_{s}	$\mathbb{Z}_2$	1 if shift $s \in \mathcal{S}$ is an administrative shift, otherwise 0, as marked in G5 in *Data sheet 2: Shift types*.
D^{mo}	$\mathbb{Z}^{0+}$	Number of days in the current month.
R^{pres}_{p}	$\mathbb{Z}^{0+}$	Maximum number of consecutive days for which one presence shift/add-on on average must be given for each physician $p \in \mathcal{P}$.
R^{alert}_{p}	$\mathbb{Z}^{0+}$	Maximum number of consecutive days for which one alert shift/add-on on average must be given for each physician $p \in \mathcal{P}$.
R^{call}_{p}	$\mathbb{Z}^{0+}$	Maximum number of consecutive days for which one on-call shift/add-on on average must be given for each physician $p \in \mathcal{P}$.
$H^{\text{endS}}_{s,a}$	$\mathbb{R}$	The amount of hours from when a shift $s \in \mathcal{S}$ and an add-on $a \in \mathcal{A}$ combination ends until midnight. If the shift-add-on-combination ends after midnight then $H^{\text{endS}}_{s,a}$ becomes the negative amount of hours from midnight until the shift ends the next day.

H_s^{startS}	$\mathbb{R}^{0+}$	The amount of hours from midnight until the start time of a shift $s \in \mathcal{S}$. The start time of a shift is assumed to always precede that of an add-on $a \in \mathcal{A}$.
S_s^{work}	$\mathbb{Z}_2$	1 if shift $s \in \mathcal{S}$ is characterized as a working shift, otherwise 0. Equivalent to G9 in *Data sheet 2: Shift types.*
D_d^{numL}	$\mathbb{Z}^{0+}$	1 if day $d \in \mathcal{D}^{\text{all}}$ is Monday, 2 if day d is Tuesday, etc.
H_p^{fDS}	$\mathbb{Z}^{0+}$	The hours equivalent to a short rest period for physician $p \in \mathcal{P}$.
H_p^{fDR}	$\mathbb{Z}^{0+}$	The hours equivalent to a regular rest period for physician $p \in \mathcal{P}$.
H_p^{rest}	$\mathbb{Z}^{0+}$	The minimum amount of hours there must be between the end of a working shift-add-on-combination until the start of the next working shift-add-on-combination for each physician $p \in \mathcal{P}$. Usually 11 hours but can vary due to local agreements.
R_p^{wRow}	$\mathbb{Z}^{0+}$	The maximum number of consecutive work days for a physician $p \in \mathcal{P}$.
S_s^{fD}	$\mathbb{Z}_2$	1 if a shift $s \in \mathcal{S}$ is part of a rest period, otherwise 0. Can only be 1 if the shift is not a working shift, i.e. $S_s^{\text{work}} = 0$. Equivalent to G8 in *Data sheet 2: Shift types.*
FD_p^{min}	$\mathbb{Z}^{0+}$	Targeted minimum number of rest periods in a schedule. Here, a short rest period is equivalent to a half rest period and a regular rest period is equivalent to one rest period.
R^{olg}	$\mathbb{Z}^{0+}$	Between 0 and 1. Is equal to the maximum fraction of all (night) presence shift/add-ons, i.e. where $RS_s^{\text{pres}} = 1$ or $RA_a^{\text{pres}} = 1$, in the schedule, which can be filled by a chief physician (OLG).
$W_{p,q}^{\text{wkd}}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ wishes to work on any day (Sat or Sun) in weekend $q \in \mathcal{Q}$, otherwise 0.
S_s^{ifWi}	$\mathbb{Z}_2$	1 if shift $s \in \mathcal{S}$ cannot be assigned unless a physician specifically wishes for it, otherwise 0 . Equivalent to G4 in *Data sheet 2: Shift types.*
S_s^{row}	$\mathbb{Z}_2$	1 if shift $s \in \mathcal{S}$ is not favorable to have in a row with another shift or add-on where S_s^{row} or A_a^{row} is also 1, respectively, otherwise 0.
A_a^{row}	$\mathbb{Z}_2$	1 if add-on $a \in \mathcal{A}$ is not favorable to have in a row with another shift or add-on where S_s^{row} or A_a^{row} is also 1, respectively, otherwise 0.

Decision variables:

$z_{p,d,s,a}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ is assigned shift $s \in \mathcal{S}$ in combination with add-on $a \in \mathcal{A}$ on day $d \in \mathcal{D}^{\text{all}}$, otherwise 0. The decision variable $x \subset z$.
$p_{p,d}^{\text{admA}}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ is assigned an administrative shift, i.e. where $S_s^{\text{adm}} = 1$, on day $d \in \mathcal{D}$ after an on-call or alert add-on $a \in \mathcal{A}$, if the physician has agreed to the possibility but does not prefer it, otherwise 0. Equivalent to' R5=C' in *Data sheet 11: Rules*.
$p_{p,d}^{\text{admS}}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ is assigned an administrative shift, $S_s^{\text{adm}} = 1$, on day $d \in \mathcal{D}$ after an on-call or alert shift $s \in \mathcal{S}$, if the physician has agreed to the possibility but does not prefer it, otherwise 0. Equivalent to' R5=C' in *Data sheet 11: Rules*.
$p_{p,d}^{\text{adm}}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ does not get an administrative shift, $S_s^{\text{adm}} = 1$, on day $d \in \mathcal{D}$ after an on-call or alert shift $s \in \mathcal{S}$ or add-on $a \in \mathcal{A}$, if the physician prefers to, otherwise 0. Equivalent to R5=P in *Data sheet 11: Rules*.
$p_{p,d}^{\text{wish}}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ does not get a wish without star-priority fulfilled on day $d \in \mathcal{D}$, otherwise 0.
p_p^{pres}	$\mathbb{Z}^{0+}$	The number of additional presence shifts/add-ons assigned to physician $p \in \mathcal{P}$ for the entire schedule, beyond the conservative maximum of $\lfloor D^{\text{mo}}/R_p^{\text{pres}} \rfloor$.
p_p^{alert}	$\mathbb{Z}^{0+}$	The number of additional alert shifts/add-ons assigned to a physician $p \in \mathcal{P}$ for the entire schedule, beyond the conservative maximum of $\lfloor D^{\text{mo}}/R_p^{\text{alert}} \rfloor$.
p_p^{call}	$\mathbb{Z}^{0+}$	The number of additional on-call shifts/add-ons assigned to a physician $p \in \mathcal{P}$ for the entire schedule, beyond the conservative maximum of $\lfloor D^{\text{mo}}/R_p^{\text{pres}} \rfloor$.
$t_{p,d}^{\text{shi}}$	$\mathbb{R}$	The hours from the end time of the shift-add-on-combination assigned on day $d \in \mathcal{D}^{\text{all}}$ until the start time of the next working shift $s \in \mathcal{S}$, i.e. where $S_s^{\text{work}} = 1$. Can be negative if a physician has has an administrative shift the day after an on-call or alert shift.
$fD_{p,d}^{\text{sho}}$	$\mathbb{Z}_2$	1 if day $d \in \mathcal{D}^{\text{all}}$ is part of a short or regular rest period, otherwise 0. Requires that the shift $s \in \mathcal{S}$ assigned on day d is a non-working shift, i.e. $S_s^{\text{work}} = 0$.
$fDH_{p,d}^{\text{sho}}$	$\mathbb{Z}_2$	1 if the amount of hours until the next working shift, $t_{p,d}^{\text{shi}}$, on day $d \in \mathcal{D}^{\text{all}}$ is $\geq H_p^{\text{fDS}}$, i.e. the minimum amount of hours required for a short free period, otherwise 0. A helper variable used to identify if day d is part of a short free period $fD_{p,d}^{\text{sho}}$.
$t_{p,d}^{\text{shiF}}$	float	The hours from the end time of the shift-add-on-combination assigned on day $d \in \mathcal{D}^{\text{all}}$ until the start time of the next shift $s \in \mathcal{S}$ which is not part of a rest period, i.e. where $S_s^{\text{fD}} = 0$. Can be negative if a physician has has an administrative shift the day after an on-call or alert shift.

$fD_{p,d}$	$\mathbb{Z}_2$	1 if day $d \in \mathcal{D}^{\text{all}}$ is part of a short or regular rest period, otherwise 0.
$fDH_{p,d}$	$\mathbb{Z}_2$	1 if the amount of hours until the next shift which is not part of a rest period, $t^{\text{shiF}}_{p,d}$, on day $d \in \mathcal{D}^{\text{all}}$ is $\geq H^{\text{fDS}}_{p}$, i.e. the minimum amount of hours required for a short free period, otherwise 0. A helper variable used to identify if day d is part of a rest period $fD_{p,d}$.
$fDH^{\text{start}}_{p,d}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ begins either a short or regular rest period on day $d \in \mathcal{D}^{\text{all}}$, otherwise 0. A helper variable used to define $fD^{\text{start}}_{p,d}$.
$fD^{\text{start}}_{p,d}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ begins either a short or regular rest period on day $d \in \mathcal{D}^{\text{all}}$ that should be counted in the current month, otherwise 0.
$fD^{\text{couS}}_{p,d}$	$\mathbb{Z}^{0+}$	The number of short rest periods, that should be counted in this month, which begins on day $d \in \mathcal{D}^{\text{all}}$ for physician $p \in \mathcal{P}$. $fD^{\text{couS}}_{p,d}$ is zero whenever $fD^{\text{start}}_{p,d}$ is zero.
$fD^{\text{couR}}_{p,d}$	$\mathbb{Z}^{0+}$	The number of regular rest periods, that should be counted in this month, which begins on day $d \in \mathcal{D}^{\text{all}}$ for physician $p \in \mathcal{P}$. $fD^{\text{couR}}_{p,d}$ is zero whenever $fD^{\text{start}}_{p,d}$ is zero.
pFD^{u}_{p}	$\mathbb{R}^{0+}$	The total amount of rest periods which physician $p \in \mathcal{P}$ is short from reaching her/his monthly minimum target of $FD^{\min}_{p}$.
$wkd^{\text{work}}_{p,q}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ works on any of the days in weekend $q \in \mathcal{Q}$, otherwise 0.
$pWkd^{\text{work}}_{p,q}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ is working during weekend $q \in \mathcal{Q}$ and also worked during the previous weekend q_{-1}, without having wished for it, otherwise 0.
p^{OLG}	float^{+}	Between 0 and 1. Is equal to the fraction of all (night) presence shifts/add-ons being covered by chief physicians beyond the target maximum fraction: R^{olg}.
$p^{\text{row}}_{p,d}$	$\mathbb{Z}_2$	1 if physician $p \in \mathcal{P}$ has a shift or an add-on on both day $d \in \mathcal{D}$ and the previous day d_{-1} which are both tagged as 'not favorable to have in a row', i.e. either $S^{\text{row}}_{s} = 1$ or $A^{\text{row}}_{a} = 1$ on both days, otherwise 0.
$p^{\text{YL}}_{p,d}$	$\mathbb{Z}_2$	1 if day $d \in \mathcal{D}^{\text{all}}$ is a Sunday where in the following week at least one of the week days Mon-Fri belong to the current month, and if physician $p \in \mathcal{P}$ belongs to the YL union, and does not get an 'off' shift, FRI_h, on either of the week days Mon-Fri following a working shift on a Sunday, otherwise 0.

3.1 Objective

The objective of the MVP model minimizes the same elements as in the *Base model*, however, additional elements are added due to some of the new constraints:

$$
\begin{aligned}
\text{Min} \quad & \sum_{p\in\mathcal{P}} (St_p^{\text{oHr}}\, h_p^{\text{ov}} + 10\, h_p^{\text{un}}) + \sum_{p\in\mathcal{P}} (30\, St_p^{\text{oHr}}\, NR_p\, h_p^{\text{ov}}) + \\
& \sum_{p\in\mathcal{P}} \sum_{c\in\mathcal{C}^{\text{perc}}} (hC_{p,c}^{\text{ov}} + 5\, hC_{p,c}^{\text{un}}) + \sum_{c\in\mathcal{C}^{\text{prio}}} p_c^{\text{prio}} + 10 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} p_{p,d}^{\text{wish}} + \\
& 10 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} p_{p,d}^{\text{adm}} + 10 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} p_{p,d}^{\text{admS}} + 10 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} p_{p,d}^{\text{admA}} + \\
& 50 \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} pWkd_{p,q}^{\text{work}} + 100 \sum_{p\in\mathcal{P}} pFD_p^{\text{u}} + 1000\, p^{\text{OLG}} + \\
& 500 \sum_{c\in\mathcal{C}} \sum_{d\in\mathcal{D}} \left(o_{c,d}^{\text{dem}} + u_{c,d}^{\text{dem}} \right) + 5 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} p_{p,d}^{\text{row}} + \\
& 20 \sum_{p\in\mathcal{P}} \left(p_p^{\text{pres}} + p_p^{\text{alert}} + p_p^{\text{call}} \right) + 10 \sum_{p\in\mathcal{P}} \sum_{d\in D^{\text{all}}} p_{p,d}^{\text{YL}}
\end{aligned}
\tag{O2}
$$

Apart from what has already been introduced in the *Base model*, the new objective elements include:

- An additional punishment to the over hours of employees who are not a normal resource, i.e. for whom $NR_p = 1$. These are substitute physicians who should only be assigned working shifts in critical situations such as when the demand cannot otherwise be fulfilled.
- A punishment, $p_{p,d}^{\text{wish}}$, for every wish without a star-priority that has not been fulfilled. See *Not fulfilled wishes* for more detail.
- A set of punishments, $p_{p,d}^{\text{adm}}$, $p_{p,d}^{\text{admS}}$, and $p_{p,d}^{\text{admA}}$, associated with the type of shift assigned after night shifts. See *The shift following a presence, alert or on-call shift* for more detail.
- A punishment, $pWkd_{p,q}^{\text{work}}$, for every time a physician works two consecutive weekends. See *Weekend work* for more detail.
- A punishment, pFD_p^{u}, for every rest period a physician is short of his or her monthly minimum target. See *Number of rest periods* for more detail.
- A punishment, p^{OLG}, associated with the percentage of all presence night shifts being covered by chief physicians (OLG). See *Fraction of chief physicians assigned to presence shifts and add-ons* for more detail.
- A punishment, $p_{p,d}^{\text{row}}$, for whenever two shifts are assigned in a row which preferably should not be. See *Avoiding successive shifts and add-on cases* for more detail.
- A set of punishments, p_p^{pres}, p_p^{alert}, and p_p^{call}, associated with being assigned too many presence, alert, and on-call shifts. See *Frequency of presence, on-call and alert shifts and add-ons* for more detail.
- A punishment, $p_{p,d}^{\text{YL}}$, for every time a physician represented by the union YL, does not get an 'off' day the Mon-Fri following a Sunday where the physician worked. See *Sunday work* for more detail.

Here, the constant factors multiplied onto each decision variable in the objective has been decided using trial and error based on; the importance of the variable, the number of times an abstract variable can occur, and the magnitude of the variable itself. As an example, it is crucial to meet all demands and a slack variable has only been added to avoid infeasibility. This slack variable therefore triggers a large punishment of 500 per violation.

3.2 Extending the model to the previous month

Multiple rules requires the scheduler to look at the shift assignments in the previous month. E.g. if a physician had a night shift on the last day, d^{-1}, of the previous month, then he or she must get a sleep shift, SOV, on the first day, d^1, of the current month. Therefore, the decision variable $z_{p,d,s,a}$ is introduced as an extension of the daily shift-add-on-assignment, $x_{p,d,s,a}$. The difference between x and z is that x only includes the days in the current month, $d \in \mathcal{D}$, whereas z includes all days in the previous and current month as well as the three first days of the coming month, $d \in \mathcal{D}^{\text{all}}$.

Firstly, in the current month, i.e. the days $d \in \mathcal{D}$, $z_{p,d,s,a}$ must be exactly equal to the schedule $x_{p,d,s,a}$:

$$z_{p,d,s,a} = x_{p,d,s,a}, \qquad \forall p \in \mathcal{P}, \forall d \in \mathcal{D}, \forall s \in \mathcal{S}, \forall a \in \mathcal{A} \tag{C2.1}$$

Secondly, since it has already been decided what the schedule looks like for the previous month, z must equal the exact combination of shifts, $OP^{\text{shi}}_{p,d}$, and add-ons, $OP^{\text{add}}_{p,d}$, which was assigned on every day d in the old plan:

$$\begin{aligned} z_{p,d,s,a} &= 1 \\ &\forall p \in \mathcal{P},\ \forall d \in \mathcal{D}^{\text{prev}},\ s = \{OP^{\text{shi}}_{p,d}\},\ a = \{OP^{\text{add}}_{p,d}\} \end{aligned} \tag{C2.2}$$

$$\begin{aligned} z_{p,d,s,a} &= 0 \\ &\forall p \in \mathcal{P},\ \forall d \in \mathcal{D}^{\text{prev}},\ s = \{OP^{\text{shi}}_{p,d}\},\ \forall a \in \mathcal{A} \backslash \{OP^{\text{add}}_{p,d}\} \end{aligned} \tag{C2.3}$$

$$\begin{aligned} z_{p,d,s,a} &= 0 \\ &\forall p \in \mathcal{P},\ \forall d \in \mathcal{D}^{\text{prev}},\ \forall s \in \mathcal{S} \backslash \{OP^{\text{shi}}_{p,d}\},\ \forall a \in \mathcal{A} \end{aligned} \tag{C2.4}$$

Lastly, for the purpose of ensuring sufficient rest periods at the end of the current month (this is done in section *Weekly rest period*), the first three days in the coming month in z, $[d^{e+1}, ..., d^{e+3}]$, are assigned the 'off' shift 'FRI_h' as well as the 'no add-on' option called 'ingen':

$$\begin{aligned} z_{p,d,s,a} &= 1 \\ &\forall p \in \mathcal{P},\ \forall d \in [d^{e+1}, ..., d^{e+4}],\ s = \{\text{FRI\_h}\},\ a = \{\text{ingen}\} \end{aligned} \tag{C2.5}$$

$$\begin{aligned} z_{p,d,s,a} &= 0 \\ &\forall p \in \mathcal{P},\ \forall d \in [d^{e+1}, ..., d^{e+4}],\ s = \{\text{FRI\_h}\},\ \forall a \in \mathcal{A} \backslash \{\text{ingen}\} \end{aligned} \tag{C2.6}$$

$$\begin{aligned} z_{p,d,s,a} &= 0 \\ &\forall p \in \mathcal{P},\ \forall d \in [d^{e+1}, ..., d^{e+4}],\ \forall s \in \mathcal{S} \backslash \{\text{FRI\_h}\},\ \forall a \in \mathcal{A} \end{aligned} \tag{C2.7}$$

3.3 Not fulfilled wishes

The purpose of the next set of constraints is to punish the objective, whenever a wish without star-priority, is not fulfilled. This can be done by modifying the hard constraints from the base model, (C1.16)-(C1.18), that ensure all star-priority wishes are fulfilled. The only modification needed is to make each of the three constraints soft and to consider wishes without a star instead of star-wishes. This can be achieved by activating a punish value, $p^{\text{wish}}_{p,d}$, on each day d that a physician p does not get her or his wish fulfilled. Furthermore, the condition on the wish priority, $WP^{\text{prio}}_{p,d}$, is now set to only consider wishes without the star-priority, whereas in (C1.16)-(C1.18) only star-priority wishes are considered:

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,d,s,a}\, WS^{\text{add}}_{w,a}\, WS^{\text{shi}}_{w,s} + p^{\text{wish}}_{p,d} = 1, \\ \forall p\in\mathcal{P},\, \forall d\in\mathcal{D},\, w=\{WP^{\text{shi}}_{p,d}\};\, WP^{\text{prio}}_{p,d} \neq \{*\} \wedge WP^{\text{shi}}_{p,d} \neq \emptyset \wedge WP^{\text{add}}_{p,d} = \emptyset \tag{C2.8}$$

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,d,s,a}\, WA^{\text{add}}_{w,a}\, WA^{\text{shi}}_{w,s} + p^{\text{wish}}_{p,d} = 1, \\ \forall p\in\mathcal{P},\, \forall d\in\mathcal{D},\, w=\{WP^{\text{add}}_{p,d}\};\, WP^{\text{prio}}_{p,d} \neq \{*\} \wedge WP^{\text{shi}}_{p,d} = \emptyset \wedge WP^{\text{add}}_{p,d} \neq \emptyset \tag{C2.9}$$

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,d,s,a}\, WS^{\text{add}}_{w_1,a}\, WS^{\text{shi}}_{w_1,s}\, WA^{\text{add}}_{w_2,a}\, WA^{\text{shi}}_{w_2,s} + p^{\text{wish}}_{p,d} = 1, \qquad \forall p\in\mathcal{P},\, \forall d\in\mathcal{D}, \\ w_1=\{WP^{\text{shi}}_{p,d}\},\; w_2=\{WP^{\text{add}}_{p,d}\};\, WP^{\text{prio}}_{p,d} \neq \{*\} \wedge WP^{\text{shi}}_{p,d} \neq \emptyset \wedge WP^{\text{add}}_{p,d} \neq \emptyset \tag{C2.10}$$

3.4 The shift following a presence, alert or on-call shift

All night shifts and night add-ons are either characterized as a presence, on-call or alert shift/add-on. Physicians are entitled to a sleep shift, SOV, on the subsequent day of any night shift. An exemption exists for on-call or alert shifts/add-ons where the physician is on stand-by from home. Since a physician does not always end up working during a stand-by shift/add-on, some physicians prefer to have an administrative shift the subsequent day and save up a free day for later. If the physician is called to work during the stand-by shift, she or he can simply push their administrative work on the subsequent day until after they wake up.
To give more flexibility to the scheduler, some physicians have also agreed to being assigned subsequent administrative shifts, if need be, but do not prefer it. Thus, the objective should be punished for all occurrence of a physician with the preference for a subsequent administrative shift, who does not get it, and similarly for physicians without the preference, who gets a subsequent administrative shift.

The first constraint ensures that if a shift is a night shift, i.e. $K^{\text{shi}}_{s,k} = 1$ where $k = \{NAT\}$, and a presence shift, $RS^{\text{pres}}_{s} = 1$, then the shift on the subsequent day must be a sleep shift, 'SOV', for all physicians. To ensure the rule is upheld on all days in all schedules, the last day in the previous month, d^{-1}, must be checked along with all but the last day in the current month:

$$\sum_{a\in\mathcal{A}} z_{p,d,s,a}\, RS^{\text{pres}}_{s} \leq \sum_{a_1\in\mathcal{A}} z_{p,d+1,s_1,a_1} \\ \forall p\in\mathcal{P}, \forall d\in[d^{-1},..,d^{e-1}], \forall s\in\mathcal{S}, s_1=\{SOV\},\, k=\{NAT\};\, K^{\text{shi}}_{s,k}=1 \tag{C2.11}$$

Similarly for a presence add-on at night. Currently the anesthesia department has no presence add-on shifts, but for scalability this option is implemented:

$$\sum_{s\in\mathcal{S}} z_{p,d,s,a}\, RA_a^{\text{pres}} \leq \sum_{a_1\in\mathcal{A}} z_{p,d+1,s_1,a_1}$$
$$\forall p \in \mathcal{P},\, \forall d \in [d^{-1}, ..., d^{e-1}],\, s_1 = \{SOV\},\, \forall a \in \mathcal{A},\, k = \{NAT\};\, K_{a,k}^{\text{add}} = 1 \tag{C2.12}$$

The next constraint ensures that if a physician has any night shift, i.e. $K_{s,k}^{\text{shi}} = 1$ where $k = \{NAT\}$, then he or she must have either a sleep shift, 'SOV', or an administrative shift s, i.e. $S_s^{\text{adm}} = 1$, on the subsequent day. Here, the physician can only be assigned an administrative shift if she or he has a preference for it or has agreed to it if needed, i.e. $R_p^{\text{pref}} = 1$ or $R_p^{\text{can}} = 1$, respectively. This constraint will only ever apply to night shifts which are on-call or alert shifts, due to (C2.11):

$$\sum_{a\in\mathcal{A}} z_{p,d,s,a} \leq \sum_{a_1\in\mathcal{A}} z_{p,d+1,s_1,a_1} + \sum_{s_2\in\mathcal{S}} \left((R_p^{\text{pref}} + R_p^{\text{can}})\, S_{s_2}^{\text{adm}} \sum_{a_2\in\mathcal{A}} z_{p,d+1,s_2,a_2} \right)$$
$$\forall p \in \mathcal{P},\, \forall s \in \mathcal{S},\, \forall d \in [d^{-1}, .., d^{e-1}],\, s_1 = \{SOV\},\, k = \{NAT\};\, K_{s,k}^{\text{shi}} = 1 \tag{C2.13}$$

Similarly for any night add-on:

$$\sum_{s\in\mathcal{S}} z_{p,d,s,a} \leq \sum_{a_1\in\mathcal{A}} z_{p,d+1,s_1,a_1} + \sum_{s_2\in\mathcal{S}} \left((R_p^{\text{pref}} + R_p^{\text{can}})\, S_{s_2}^{\text{adm}} \sum_{a_2\in\mathcal{A}} z_{p,d+1,s_2,a_2} \right)$$
$$\forall p \in \mathcal{P},\, \forall a \in \mathcal{A},\, \forall d \in [d^{-1}, .., d^{e-1}],\, s_1 = \{SOV\},\, k = \{NAT\};\, K_{a,k}^{\text{add}} = 1 \tag{C2.14}$$

The next constraint ensures a punishment, $p_{p,d}^{\text{admS}}$, is added to the objective for every instance of a physician getting an administrative shift on the subsequent day of a night shift s, in the case that the physician has allowed this to happen but does not prefer it, i.e. denoted by $R_p^{\text{can}} = 1$:

$$\sum_{a\in\mathcal{A}} z_{p,d,s,a}\, R_p^{\text{can}} \leq \sum_{a_1\in\mathcal{A}} z_{p,d+1,s_1,a_1} + p_{p,d+1}^{\text{admS}}$$
$$\forall p \in \mathcal{P}, \forall s \in \mathcal{S}, \forall d \in [d^{-1}, .., d^{e-1}],\, s_1 = \{SOV\},\, k = \{NAT\};\, K_{s,k}^{\text{shi}} = 1 \tag{C2.15}$$

Similarly, a punishment, $p_{p,d}^{\text{admA}}$, is added to the objective for the subsequent days of every night add-ons a:

$$\sum_{s\in\mathcal{S}} z_{p,d,s,a}\, R_p^{\text{can}} \leq \sum_{a_1\in\mathcal{A}} z_{p,d+1,s_1,a_1} + p_{p,d+1}^{\text{admA}}$$
$$\forall p \in \mathcal{P}, \forall a \in \mathcal{A}, \forall d \in [d^{-1}, .., d^{e-1}],\, s_1 = \{SOV\},\, k = \{NAT\};\, K_{a,k}^{\text{add}} = 1 \tag{C2.16}$$

The last constraint ensures that a similar punish value, $p_{p,d}^{\text{adm}}$, is added to the objective for every instance of a physician getting a sleep shift, 'SOV', on the subsequent day of a night shift s or night add-on a, if the physician would have preferred to be assigned an administrative shift instead, i.e. $R_p^{\text{pref}} = 1$. If the night shift/add-on is a presence shift/add-on, $RS_s^{\text{pres}} = 1$ or $RA_a^{\text{pres}} = 1$, respectively, then the punish value will not be activated, since the physician then *must* be assigned a sleep shift.

$$\begin{aligned}\sum_{a\in\mathcal{A}} z_{p,d+1,s,a}\, R_p^{\text{pref}} \le p_{p,d+1}^{\text{adm}} + \sum_{s_1\in\mathcal{S}}\sum_{a_1\in\mathcal{A}} z_{p,d,s_1,a_1}(RS_{s_1}^{\text{pres}} + RA_{a_1}^{\text{pres}})\\ \forall p\in\mathcal{P}, \forall d\in[d^{-1},..,d^{e-1}],\ s=\{SOV\}\end{aligned} \tag{C2.17}$$

3.5 Frequency of presence, on-call and alert shifts and add-ons

All shifts and add-ons occurring outside regular working hours (6-18), are either categorized as a; presence, alert, and on-call shift/add-on. Each of these shift types can at most occur every 'x' days on average over a norm period as explained in R6, R7, and R8 in *Data sheet 11: Rules*. Here, a norm period is usually between 3-12 months, depending on the union of each physician. A norm period can start and end on any day in a month.
To avoid having to consider an entire norm period, a conservative upper limit on the number of assigned presence, alert, and on-call shifts/add-ons, is set. If the upper limit is breached, a punish value, p_p^{pres}, p_p^{pres}, and p_p^{pres}, respectively, is added for every shift assigned above the limit. The upper limit is determined by flooring the number of days in the current month, D^{mo}, divided by the maximum occurrence rate of presence, alert, and on-call shifts/add-ons, respectively. E.g. a presence shift/add-on can under regular circumstances occur every six days on average, thus, in a month with 31 days, the objective will be punished if more than $\lfloor 31/6\rfloor = 5$ presence shifts/add-ons are assigned to a physician.
The binary indicators for whether a shift or an add-on are categorized as a presence shift or add-on, RS_s^{pres} and RS_a^{pres}, respectively, are used to count the number of occurrences of all presence shifts/add-ons:

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,s,a}^{\text{help}}(RS_s^{\text{pres}} + RA_a^{\text{pres}}) \le \left\lfloor \frac{D^{\text{mo}}}{R_p^{\text{pres}}}\right\rfloor + p_p^{\text{pres}}, \qquad \forall p\in\mathcal{P} \tag{C2.18}$$

Similarly for alert and on-call shifts/add-ons, respectively:

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,s,a}^{\text{help}}(RS_s^{\text{alert}} + RA_a^{\text{alert}}) \le \left\lfloor \frac{D^{\text{mo}}}{R_p^{\text{alert}}}\right\rfloor + p_p^{\text{alert}}, \qquad \forall p\in\mathcal{P} \tag{C2.19}$$

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,s,a}^{\text{help}}(RS_s^{\text{call}} + RA_a^{\text{call}}) \le \left\lfloor \frac{D^{\text{mo}}}{R_p^{\text{call}}}\right\rfloor + p_p^{\text{call}}, \qquad \forall p\in\mathcal{P} \tag{C2.20}$$

3.6 Time between working shifts

The Danish Working Time Directive requires a minimum of 11 hours of rest between the end time of a working shift until the beginning of the next working shift (R1 in *Data sheet 11: Rules*). Furthermore, a physician must have at least one short rest period every week (R3 in *Data sheet 11: Rules*), and no more than six days must be between two successive short or regular rest periods (R2 in *Data sheet 11: Rules*).
To ensure these regulations, it is essential to know the rest time between all successive

working shifts, $t^{\text{shi}}_{p,d}$. The upper bound on the rest time to the next working shift on each day can be expressed as:

$$t^{\text{shi}}_{p,d} \leq \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left(H^{\text{endS}}_{s,a} \, z_{p,d,s,a} + H^{\text{startS}}_{s} \, z_{p,d_1,s,a} \right) +$$
$$24(d_1 - d - 1) + 24 \cdot 4 \sum_{s_1 \in \mathcal{S}} \sum_{a_1 \in \mathcal{A}} \left(1 - S^{\text{work}}_{s_1} \right) z_{p,d_1,s_1,a_1} \tag{C2.21}$$
$$\forall p \in \mathcal{P},\ d \in [d^{-R^{\text{wRow}}_p}, ..., d^e],\ d_1 \in [d+1, ..., d+3]$$

Let us consider the time between the end time of a shift s and add-on a combination on day d and the start time of a shift s_1 and add-on a_1 combination on day d_1, where day d_1 is a successor of day d. Then the rest time between the two shift-add-on combinations can be calculated as the sum of the following three contributions, which are pictured in Figure 3.2:

1. The amount of hours from the end of the shift s and add-on a combination on day d, until midnight 00.00 the same day. E.g. if the s and a combination ends at 16:00, then this contribution, $H^{\text{endS}}_{s,a}$, becomes 8 hours. If the s and a combination ends after midnight, e.g. at 09.00 the next morning, then the contribution, $H^{\text{endS}}_{s,a}$, becomes -9 hours.

2. The number of days between d and d_1, d and d_1 excluded, multiplied by 24 hours. E.g. if d is the 20^{th} and d_1 the 23^{rd}, then there are two days in between, the 21^{st} and the 22^{nd}. Thus the contribution becomes: $24(d_1 - d - 1) = 24 \cdot 2 = 48$ hours.

3. The amount of hours from midnight 00.00 until the start time of the shift s_1 on day d_1. Here, it is assumed that an add-on always begins at the same time or later than the shift it is assigned with, thus, it suffices to consider the start time of the shift. If s_1 starts at e.g. 07:45, then the contribution, $H^{\text{startS}}_{s_1}$, becomes: 7.75 hours.

The above corresponds to the first three terms in (C2.21).
As pictured in Figure 3.2, the idea is to find the time from the shift-add-on-combination on day d to the shift-add-on-combination on all the following days; d^{+1}, d^{+2}, ..., d^e. As we wish to find the time until the next working shift s_1, thereby determining the duration of the rest period, we are not interested in finding the time from day d to a free day, i.e. a day with a shift such as FRI (off) or FE (vacation). For the sake of creating an upper bound for the time to the next working shift, $t^{\text{shi}}_{p,d}$, we therefore add a large amount of

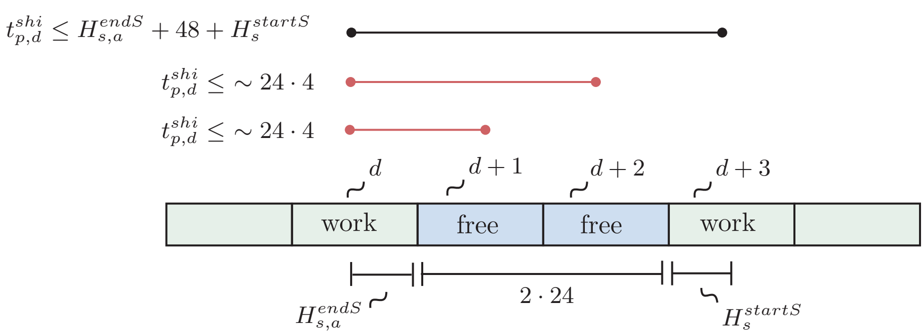

Figure 3.2: Illustration of how $ts^{\text{shi}}_{p,s}$ is upper bounded in (C2.21). Here, the constraint which acts as the true upper bound is illustrated with a black line and the two other constraints, marked with red lines, are both less tight than the true upper bound. This is the intention since day $d+1$ and $d+2$ are both free days and $ts^{\text{shi}}_{p,s}$ should represent the time to the next working shift.

hours to the three-step calculation of the rest time, whenever a shift s_1 on a successive day is not a working shift, i.e. $S^{\text{work}}_{s_1} = 0$. Here, a large amount of hours is chosen to be; 24 hours times four days. Why this amount is large enough is explained later. This corresponds to the last term in (C2.21).
The time to the next working shift, $t^{\text{shi}}_{p,d}$, on day d, must then be smaller than or equal to the rest time calculation to all of the successive days. This way, it will be the time to the first working shift that will upper bound the value of $t^{\text{shi}}_{p,d}$ as seen in Figure 3.2.

In terms of the 11 hours rest rule and ensuring, as a minimum, a short rest period (currently 32 hours) every week, it does not matter if the time to the next working shift is 32 hours or 150 hours. This fact can be utilized to avoid generating unnecessary constraints. If a working shift on day d is followed by three 'off' shifts, then even if the working shift on day d is a night shifts ending the next morning, the rest period will still be more than 48 hours long. Thus, on every day d, it is only necessary to calculate the time to the shifts assigned on each of the three subsequent days. This is why 24 times four days is a sufficiently large amount of hours to add to 'off' days, as the time to any working shift within the first three days will always be less than this amount. Moreover, to ensure that physicians who worked the last six days of the past month, are assigned at least a short rest period in the beginning of the current month, it is also necessary to consider the last R^{wRow}_p = six days of the previous schedule.
Lastly, the first three days of the coming month, d^{e+1} to d^{e+3}, are also considered when upper bounding $t^{\text{shi}}_{p,d}$ in (C2.21). By fixing the three first days in the coming month to 'off' shift, as done in (C2.5) - (C2.7), it is always possible to get a short rest period at the end of the current month.

The lower bound on the rest time to the next working shift on each day can be expressed as:

$$
\begin{aligned}
t^{\text{shi}}_{p,d} \geq \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} & \left(H^{\text{endS}}_{s,a}\ z_{p,d,s,a} + H^{\text{startS}}_{s}\ z_{p,d_1,s,a}\right) + \\
& 24\,(d_1 - d - 1) - 24\cdot 4 \sum_{s_1\in\mathcal{S}}\sum_{a_1\in\mathcal{A}} \left(1 - S^{\text{work}}_{s_1}\right) z_{p,d_1,s_1,a_1} + \\
& 24\cdot 4 \left(1 - \sum_{d_2=d+1}^{d_1}\sum_{s_2\in\mathcal{S}}\sum_{a_2\in\mathcal{A}} S^{\text{work}}_{s_2}\ z_{p,d_1,s_2,a_2} - \sum_{s_3\in\mathcal{S}}\sum_{a_3\in\mathcal{A}} (1 - S^{\text{work}}_{s_3})\ z_{p,d_2,s_3,a_3}\right) \\
& \qquad \forall p \in \mathcal{P},\ d \in [d^{-R^{\text{wRow}}_p}, ..., d^{e}],\ d_1 \in [d+1, ..., d+3]
\end{aligned}
\tag{C2.22}
$$

When lower bounding the rest time until the next working shift, a similar but opposite approach is taken. Again, the rest time until the shift on each of the consecutive days is calculated with the three contribution sum approach, and again, the rest time up until an 'off' shift should not be considered. As opposed to the upper bound constraint, (C2.21), a large amount of hours are now subtracted instead of added whenever considering the rest time up until an 'off' shift. This corresponds to the fourth term in (C2.22). For the same reason as before, the large amount of hours is chosen as 24 hours times four days.
For the lower bound, consecutive work days are now a problem. We wish for the rest time up until the first working shift to act as the tightest lower bound on $t^{\text{shi}}_{p,d}$, thus the rest time calculated up until any other day must be less as pictured in Figure 3.3. Here, if both day $d+2$ and day $d+3$ are work days, then the time up until the shift on day $d+3$ is greater than the time up until the shift on day $d+2$, when using the three-part sum approach.

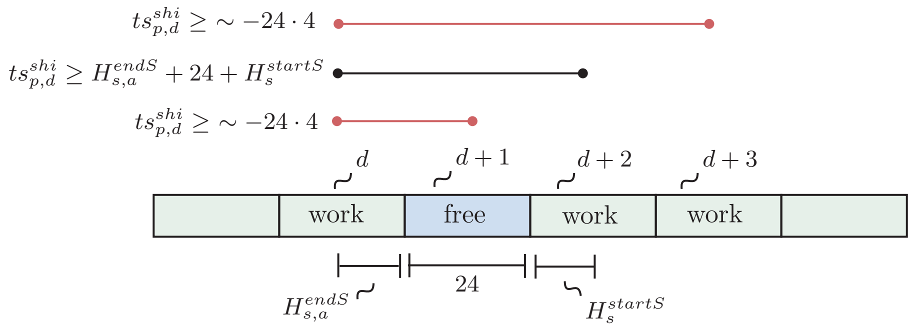

Figure 3.3: Illustration of how $ts^{\text{shi}}_{p,s}$ is lower bounded in (C2.22). Here, the constraint which acts as the true upper bound is illustrated with a black line and the two other constraints, marked with red lines, are both less tight than the lower upper bound. This is the intention since day $d+1$ is a free day and $d+3$ is the second work shift after day d and $ts^{\text{shi}}_{p,s}$ should represent the time to the next working shift.

Thus, when considering a span of days containing more than one work day, a large amount of hours must be subtracted from all but the first subsequent work day. This is done by summing up over all the work shifts in the span of days considered from $d+1$ to $d+i$, $i \leq 3$, subtracting this amount from 1, and then multiplying the resultant number by 24 hours times 4 days, i.e. a sufficiently large number. This corresponds to the fifth term in (C2.22). By adding this amount to the rest time calculation, nothing will be added when the span of days contains exactly one work shift because $1-1$ cancels the contribution. When the span of days contains two or more work days, then 1 minus the number of work days will be a negative number and as a result a large negative contribution will be added to the rest time calculation. The only flaw with this approach is when the span of days contains no work shifts, since 1 minus zero work shifts will add a positive contribution to the rest time calculation. This is avoided by subtracting $(1 - S^{\text{work}}_{s})$, where the shift s is the shift on day $d+i$. Thus, if this shift is an 'off' shift, the positive contribution cancels. This corresponds to the sixth and last term in (C2.22). A small example of how the regulating terms in the lower bound (C2.22) work, is shown in Figure 3.4.

In conclusion, the upper and lower bound on $t^{\text{shi}}_{p,d}$, (C2.21) and (C2.22), will be exactly equal each other and most tight on the first subsequent day d_1 which is a working day. Therefore, $t^{\text{shi}}_{p,d}$ will be exactly equal to the rest time after the shift-add-on-combination assigned on day d. The only exception is if day d is followed by at least three 'off' shifts, in which case $t^{\text{shi}}_{p,d}$ will be lower bounded by $\sim -24 \cdot 4$ and upper bounded by $\sim 24 \cdot 4$. This is fine, since the solver can then freely choose the rest period, $t^{\text{shi}}_{p,d}$, to be 35 hours long if

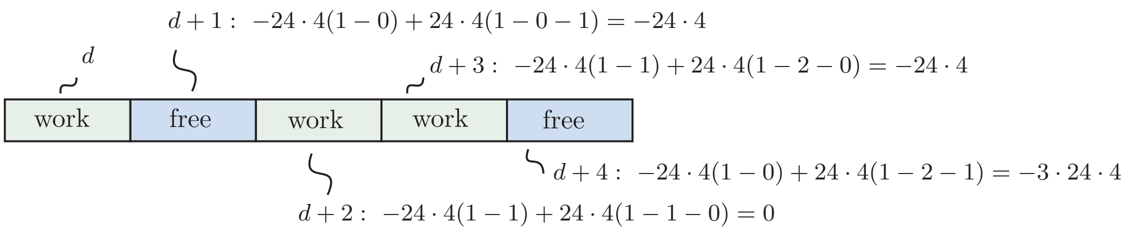

Figure 3.4: An example of how the regulating terms in (C2.22), the lower bound of $ts^{\text{shi}}_{p,s}$, works. Here, the calculations shown corresponds to all but the first three terms in (C2.22). As can be seen, the regulating terms become a large negative number on all days except the first work day after day d.

a short rest period is needed to oblige by the Danish Working Time Directive.

$t^{\text{shi}}_{p,d}$ is defined for all of the days in the current and in the previous month. As only the last R^{wRow}_p number of days in the previous month are bounded in (C2.21) and (C2.22), the remaining days are fixed to ease the workload of the solver. Here, it is assumed that the previous plan is compliant with the Danish Working Time Directive, thus, on all remaining days, $t^{\text{shi}}_{p,d}$ is fixed to the amount of hours in a short rest period, H^{fDS}_p:

$$t^{\text{shi}}_{p,d} = H^{\text{fDS}}_p \qquad \forall p \in \mathcal{P},\ d \in [d^s, ..., d^{-R^{\text{wRow}}_p - 1}] \tag{C2.23}$$

3.7 The 11 hour rule

Now that the time to the next working shift, $t^{\text{shi}}_{p,d}$, is known for each day d in the schedule, the 11 hour rule enforced by the Danish Working Time Directive can readily be formulated mathematically:

$$\begin{aligned} t^{\text{shi}}_{p,d} \geq H^{\text{rest}}_p - 2H^{\text{rest}}_p \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \Big(z_{p,d,s,a} \Big(R^{pref}_p + R^{can}_p \Big) \\ \Big(RA^{\text{alert}}_a + RA^{\text{call}}_a + RS^{\text{alert}}_s + RS^{\text{call}}_s \Big) \Big) - 2H^{\text{rest}}_p \sum_{s_1 \in \mathcal{S}} (z_{p,d,s_1,a_1}) \\ \forall p \in \mathcal{P},\ d \in [d^{-1}, ..., d^{e-1}],\ a_1 = \{AKC\} \end{aligned} \tag{C2.24}$$

This constraint consists of three parts. The first part ensures that the hours until the next shift must be at least H^{rest}_p, which is typically 11 hours.
The second part describes the first exception to this rule, namely when a physician has agreed to the possibility of having an administrative shift the day after an on-call or alert shift/add-on, i.e. $R^{pref}_p \vee R^{can}_p = 1$. In this case, there will not be sufficient rest time and the constraint is turned 'off' by subtracting $2H^{\text{rest}}_p$. Note that (C2.11)-(C2.17) still ensures that no other shifts than a sleep shift, 'SOV', or an administrative shift can be assigned the day after an on-call or alert shift/add-on.
The third part describes the third exception to the 11-hour rule. This exception is unique to the anesthesia department and thus would not scale to other departments within Rigshospitalet. However, with a binary indicator for each shift and each add-on, it would be possible to make this exception generic. The exception concerns the add-on 'AKC' which occurs during the night. A physician assigned 'AKC' can only be called on the phone and does not ever need to go to the clinic to work. A local agreement has therefore been made such that 11 hours of rest is not required after an 'AKC' add-on. Thus, in the case of an 'AKC' add-on assignment there will not be sufficient rest time if the following day is also a work day and the constraint is turned 'off' by subtracting $2H^{\text{rest}}_p$.

3.8 Weekly rest period

This set of constraints ensure that every physician gets at least one short rest period every week (R3 in *Data sheet 11: Rules*), and that there are no more than six days between two successive short or regular rest periods (R2 in *Data sheet 11: Rules*), as required by Danish Working Time Directive. Table 3.1 illustrates an example of how the majority of the constraints in this section works and is useful to consolidate while reading.

$fDH^{\text{sho}}_{p,d}$	$t^{\text{shi}}_{p,d}$	$> H^{\text{fDS}}_p$	$> H^{\text{fDS}}_p$	$> H^{\text{fDS}}_p$	$< H^{\text{fDS}}_p$	$< H^{\text{fDS}}_p$	$< H^{\text{fDS}}_p$	$< H^{\text{fDS}}_p$
	(C2.25-C2.26):	1	1	1	0	0	0	0
		work	free	free	free	work	free	work
$fD^{\text{sho}}_{p,d}$	(C2.27):	0				0		0
	(C2.28):		1	1	(1)			
	(C2.30):			1	1			
	(C2.31):						1	

Table 3.1: Illustration of how $fD^{\text{sho}}_{p,d}$ (one if day d is part of a sufficiently long rest period) is bounded using (C2.25)-(C2.31). In the above, only the tight bounds are shown. The example shows that $fD^{\text{sho}}_{p,d}$ is tightly bounded every day by at least one of the constraints.

Initially, a helper variable must be defined for later use. The helper variable, $fDH^{\text{sho}}_{p,d}$, must be equal to one when the time to the next working shift on day d, $t^{\text{shi}}_{p,d}$, is at least equal to the amount of rest time required for a short rest period, H^{fDS}_p, otherwise zero. First, it is ensured that $fDH^{\text{sho}}_{p,d}$ is equal to one, whenever $t^{\text{shi}}_{p,d} \geq H^{\text{fDS}}_p$. Here, 24 hours times the number of days in the current month, D^{mo}, is a sufficiently large number for the constraint to work, as there can never be that much time between two shifts in the same schedule:

$$t^{\text{shi}}_{p,d} - H^{\text{fDS}}_p < 24\, D^{\text{mo}}\, fDH^{\text{sho}}_{p,d}, \qquad \forall p \in \mathcal{P},\ d \in [d^s, ..., d^{e-1}] \tag{C2.25}$$

Next, it is ensured that $fDH^{\text{sho}}_{p,d}$ is equal to zero, whenever $t^{\text{shi}}_{p,d} < H^{\text{fDS}}_p$:

$$H^{\text{fDS}}_p - t^{\text{shi}}_{p,d} \leq 24\, D^{\text{mo}}(1 - fDH^{\text{sho}}_{p,d}), \qquad \forall p \in \mathcal{P},\ d \in [d^s, ..., d^{e-1}] \tag{C2.26}$$

Another binary decision variable needed, is an indicator for which days d in this schedule and in the schedule of the past month, are part of, as a minimum, a short rest period for physician p. If the shift given on day d is not a working shift and if day d is part of a rest period of at least H^{fDS}_p hours, then $fD^{\text{sho}}_{p,d}$ must be one, otherwise zero. Thus, the first constraint forces $fD^{\text{sho}}_{p,d}$ to be zero on all days where a work shift is given, i.e. where $S^{\text{work}}_s = 1$:

$$fD^{\text{sho}}_{p,d} \leq 1 - \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} S^{\text{work}}_s\, z_{p,d,s,a} \qquad \forall p \in \mathcal{P},\ d \in [d^s, ..., d^e] \tag{C2.27}$$

Secondly, as a minimum, the first 'off' day in a rest period with a duration of at least H^{fDS}_p hours, is forced to be one:

$$fDH^{\text{sho}}_{p,d-1} + \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} (1 - S^{\text{work}}_s)\, z_{p,d,s,a} \leq 1 + fD^{\text{sho}}_{p,d} \qquad \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^e] \tag{C2.28}$$

This will only happen when both of the following conditions are true:

1. $fDH^{\text{sho}}_{p,d-1}$ is one on the previous day, $d-1$, i.e. the time to the next working shift was yesterday more than H^{fDS}_p hours.

2. The shift assigned on the following day d, is an 'off' shift.

For all other cases than the one described above, (C2.28) allows $fD^{\text{sho}}_{p,d}$ to be either one or zero.
(C2.28) cannot bind $fD^{\text{sho}}_{p,d}$ on the very first day of the previous month, since the time to the next working shift is not known for the previous day. Therefore, $fD^{\text{sho}}_{p,d}$ is bound to one if it is an 'off' day, otherwise zero on this day:

$$fD^{\text{sho}}_{p,d} = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}(1 - S^{\text{work}}_s)\, z_{p,d,s,a} \qquad \forall p \in \mathcal{P},\ d = \{1\} \tag{C2.29}$$

Thirdly, all but the first off-day in a rest period with a duration of at least H^{fDS}_p hours, are forced be one:

$$\begin{gathered} fD^{\text{sho}}_{p,d-1} + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}(1 - S^{\text{work}}_s)\, z_{p,d,s,a} \leq\ 1 + fD^{\text{sho}}_{p,d} \\ \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^e] \end{gathered} \tag{C2.30}$$

This will only happen when both of the following conditions are true:

1. $fD^{\text{sho}}_{p,d-1}$ is one on the previous day, $d-1$, i.e. the previous day is part of a rest period with a duration of at least H^{fDS}_p hours.
2. The shift assigned on the following day d, is an 'off' shift.

For all other cases than the one described above, (C2.30) allows $fD^{\text{sho}}_{p,d}$ to be either one or zero.

The last thing to ensure, is that 'off' days which are not part of a sufficiently long rest period, are forced to be zero:

$$\begin{gathered} fDH^{\text{sho}}_{p,d-1} + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\left(S^{\text{work}}_s\, z_{p,d,s,a}\right) + fD^{\text{sho}}_{p,d-1} \geq\ fD^{\text{sho}}_{p,d} \\ \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^e] \end{gathered} \tag{C2.31}$$

This will happen when all of the following conditions are true:

1. $fDH^{\text{sho}}_{p,d-1}$ is zero on the previous day, $d-1$, i.e. on the previous day the time to the next work shifts is less than H^{fDS}_p hours.
2. The shift assigned on the following day d, is an 'off' shift, i.e. $S^{\text{work}}_s = 0$.
3. $fD^{\text{sho}}_{p,d-1}$ is zero on the previous day, $d-1$, i.e. the previous day is not part of a sufficiently long rest period.

For all other cases than the one described above, where day d is an 'off' shift, $fD^{\text{sho}}_{p,d}$ is not bounded. Since $fD^{\text{sho}}_{p,d}$ is already forced to be zero on all work days by (C2.27), it is irrelevant whether the above constraint is a proper upper bound for work days.

All variables needed to ensure that there are at most $R^{\text{wRow}}_p = 6$ days in between two sufficiently long rest periods, have now been defined. As the binary variable $fD^{\text{sho}}_{p,d}$ is one on days d that are part of such a rest period, one can simply check that for every $R^{\text{wRow}}_p + 1 = 7$ subsequent days, the sum of $fD^{\text{sho}}_{p,d}$ is at least one:

$$\sum_{d_1=d}^{d+R^{\text{wRow}}_p} fD^{\text{sho}}_{p,d_1} \geq 1 \qquad \forall p \in \mathcal{P},\ d \in [d^{-R^{\text{wRow}}_p}, ..., d^{e-R^{\text{wRow}}_p}] \tag{C2.32}$$

The only exception to the above is if a rest period begins with the the sleep shift 'SOV' and a physician has worked the past $R_p^{\text{wRow}} = 6$ days before the sleep shift. A sleep shift is only assigned when the shift on the previous day extends into the night, i.e. past midnight. This means that a physician which is assigned a sleep shift on the 7^{th} day after six work days ends up working one day too much. Thus, an additional constraint is added for this specific case:

$$\sum_{d_2=d}^{d+R_p^{\text{wRow}}} fD_{p,d_2}^{\text{sho}} - \sum_{a \in \mathcal{A}} z_{p,d_1,s_1,a} + (1 - fD_{p,d_1}^{\text{sho}}) \geq 1 \tag{C2.33}$$
$$\forall p \in \mathcal{P},\, d \in [d^{-R_p^{\text{wRow}}}, ..., d^{e-R_p^{\text{wRow}}}],\, d_1 = d + R_p^{\text{wRow}},\, s_1 = \{SOV\}$$

From the above one can deduce that if on the 7^{th} day ($d_1 = d + R_p^{\text{wRow}}$), physician p is assigned a sleep shift and this sleep shift is part of a sufficiently long rest period, i.e. $fD_{p,d_1}^{\text{sho}} = 1$, then there must be at least one other day that is also part of rest period in the seven sequential days considered.
A more scalable approach to this exception could be to utilize a binary identifier which would be one for any shift or add-on which extends past midnight, otherwise zero. Thus, (C2.33) could be made more generic.

3.9 Number of rest periods

Over the course of a norm period, a physician has the right to a minimum number of rest periods. Here, a regular rest period with a duration of H_p^{fDR} hours counts as one rest period and two short rest periods with a duration of H_p^{fDS} hours each, also count as one rest period. Thus, for simplicity a short rest period counts as half a rest period.
Just like in the above section, *Weekly rest period*, the time to the next working shift must be known such that based on this rest time, the number of rest periods can be counted. However, the previous rest time calculations cannot be used here. This can be illustrated with a simple example:

1. A physician works six days in a row, Tuesday-Sunday, followed by a one week vacation from Monday-Sunday.
   (a) Here, despite the physician being 'off' whilst on vacation, the entire vacation period should not count towards his or her number of rest periods, as a one week vacation easily amounts to the total number of rest periods usually given in an entire month. A physician should of course not be assigned less rest periods in a month with a vacation period. On the other hand, the weekend in a vacation period should still count towards the number of entitled rest periods, as the physician will otherwise be approximately two rest periods short after a two week holiday. Thus, more rest periods should of course not be assigned after being back from a vacation period either.
   (b) In terms of ensuring that at least a short rest period is given after at most six days of work, this example is in line with the rules. It does of course not make sense, to have to assign 1-2 'off' shifts before a vacation period, to ensure sufficient rest prior to being on vacation.

In section *Time between working shifts* and *Weekly rest period*, all 'off' shifts count towards a rest period, whereas in this section, some 'off' shifts should not count towards rest periods. This includes, to name a few shifts; FE_h (vacation on a week day), OM (child care day/'omsorgsdag'), ASP (compensatory leave/'afspadsering'), etc.

Most of the constraints in this section are therefore the same as in *Time between working shifts* and *Weekly rest period*. The only difference is that wherever all 'off' shifts where considered before by; $\left(1 - S_s^{\text{work}}\right)$, now, shifts for which $S_s^{\text{fD}} = 1$ are considered instead. Here, S_s^{fD} is one for all shifts s that can be part of a rest period which should be counted, otherwise zero.

The following constraint is an upper bound for $t_{p,d}^{\text{shiF}}$ and is equivalent to (C2.21). Here, $t_{p,d}^{\text{shiF}}$ replaces $t_{p,d}^{\text{shi}}$ as the rest time until the next working shift begins:

$$\begin{aligned} t_{p,d}^{\text{shiF}} \leq & \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left(H_{s,a}^{\text{endS}}\, z_{p,d,s,a} + H_s^{\text{startS}}\, z_{p,d_1,s,a} \right) + \\ & 24(d_1 - d - 1) + 24 \cdot 15 \sum_{s_1 \in \mathcal{S}} \sum_{a_1 \in \mathcal{A}} S_{s_1}^{\text{fD}}\, z_{p,d_1,s_1,a_1} \\ & \forall p \in \mathcal{P},\ d \in [d^{-13}, ..., d^{e-1}],\ d_1 \in [d+1, ..., d+14];\ d_1 \leq d^{e+3} \end{aligned} \tag{C2.34}$$

Before, it was enough to calculate the time up until the start time of the shifts assigned on the subsequent three days. This was because it was only necessary to determine if the duration of a rest period was above a certain threshold. Now however, for each rest period the exact duration must be known to accurately determine how many short and regular rest periods the duration of the rest period corresponds to. It is assumed that no rest period is more than 14 days long, thus, the time calculation is on all days d performed on the subsequent 14 days. If a physician is 'off' for longer than two week then she or he must either be on vacation, sick leave, compensatory leave, or similar, not simply 'off'.

The next constraint is a lower bound for $t_{p,d}^{\text{shiF}}$ and is equivalent to (C2.22):

$$\begin{aligned} t_{p,d}^{\text{shiF}} \geq & \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left(H_{s,a}^{\text{endS}}\, z_{p,d,s,a} + H_s^{\text{startS}}\, z_{p,d_1,s,a} \right) + \\ & 24\,(d_1 - d - 1) - 24 \cdot 15 \sum_{s_1 \in \mathcal{S}} \sum_{a_1 \in \mathcal{A}} S_{s_1}^{\text{fD}}\, z_{p,d_1,s_1,a_1} + \\ & 24 \cdot 15 \left(1 - \sum_{d_2 = d+1}^{d_1} \sum_{s_2 \in \mathcal{S}} \sum_{a_2 \in \mathcal{A}} \left(1 - S_{s_2}^{\text{fD}} \right)\ z_{p,d_1,s_2,a_2} - \sum_{s_3 \in \mathcal{S}} \sum_{a_3 \in \mathcal{A}} S_{s_3}^{\text{fD}}\, z_{p,d_2,s_3,a_3} \right) \\ & \forall p \in \mathcal{P},\ d \in [d^{-13}, ..., d^{e-1}],\ d_1 \in [d+1, ..., d+14];\ d_1 \leq d^{e+3} \end{aligned} \tag{C2.35}$$

The only case in which the above upper and lower bound, (C2.34) and (C2.35), leaves $t_{p,d}^{\text{shiF}}$ practically unbounded is if a physician ends the current schedule with one or more consecutive 'off' shifts which all counts towards a rest period. Here, the time to the next working shift will be lower bounded by $-24 \cdot 15$ and upper bounded by $24 \cdot 15$ on the last working day of the schedule. This is not an issue since a rest period which does not end within the current schedule, consequently will not counted in the current month. A rest period can only be counted once it has ended, and thus its full duration can be determined.

As in *Weekly rest period*, the helper variable $fDH_{p,d}^{\text{sho}}$, must also be introduced. This variable will be reintroduced as $fDH_{p,d}$. The following two constraints are equivalent with (C2.25) and (C2.26), respectively:

$$t_{p,d}^{\text{shiF}} - H_p^{\text{fDS}} < 24\, D^{\text{mo}}\, fDH_{p,d}, \qquad \forall p \in \mathcal{P},\ d \in [d^s, ..., d^{e-1}] \tag{C2.36}$$

$$H_p^{\text{fDS}} - t_{p,d}^{\text{shiF}} \leq 24\, D^{\text{mo}} (1 - fDH_{p,d}), \qquad \forall p \in \mathcal{P},\ d \in [d^s, ..., d^{e-1}] \tag{C2.37}$$

Furthermore, the binary decision variable $fD^{\text{sho}}_{p,d}$ fro *Weekly rest period* is also reintroduced as $fD_{p,d}$. Again, $fD_{p,d}$ is an indicator for which days d in this schedule and in the schedule of the past month, are part of, as a minimum, a short rest period for physician p. The following five constraints are equivalent with (C2.27)-(C2.31), respectively:

$$fD_{p,d} \leq 1 - \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} (1 - S^{\text{fD}}_s)\, z_{p,d,s,a} \qquad \forall p \in \mathcal{P},\ d \in [d^s, ..., d^e] \tag{C2.38}$$

$$fDH_{p,d-1} + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} S^{\text{fD}}_s\, z_{p,d,s,a} \leq 1 + fD_{p,d} \qquad \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^e] \tag{C2.39}$$

$$fD_{p,d} = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} S^{\text{fD}}_s\, z_{p,d,s,a} \qquad \forall p \in \mathcal{P},\ d = \{1\} \tag{C2.40}$$

$$fD_{p,d-1} + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} S^{\text{fD}}_s\, z_{p,d,s,a} \leq 1 + fD_{p,d} \qquad \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^e] \tag{C2.41}$$

$$fDH_{p,d-1} + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \left(1 - S^{\text{fD}}_s\right) z_{p,d,s,a} + fD_{p,d-1} \geq fD_{p,d} \qquad \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^e] \tag{C2.42}$$

At this point, the current section, *Number of rest periods*, starts to differ from *Weekly rest period*. The idea is to calculate how many short and regular rest periods every rest period in the schedule corresponds to, as pictured in Figure 3.5. Here, the number of regular rest periods assigned are firstly maximized and then, based on the remaining hours, a short rest period is assigned as well if possible. The number of both regular and short rest periods are always assigned to the first day in a rest period. Thus, a new binary decision variable, $fDH^{\text{start}}_{p,d}$, is needed. $fDH^{\text{start}}_{p,d}$ is a helper variable and should be one on the first day of all rest periods with a duration that is equivalent to a short rest period or longer, otherwise zero. $fDH^{\text{start}}_{p,d}$ is defined for all days in the current and past month.
This problem is equivalent to the set of startup/shutdown constraints widely used in literature, perhaps most commonly for switching machines on and off. Here, the intention is similarly to turn a rest period on. Firstly, the first day of the past month is initialized using the value of $fD_{p,d}$ which is one if a day d is part of a sufficiently long rest period:

$$fDH^{\text{start}}_{p,d} = fD_{p,d}, \qquad \forall p \in \mathcal{P},\ d = \{1\} \tag{C2.43}$$

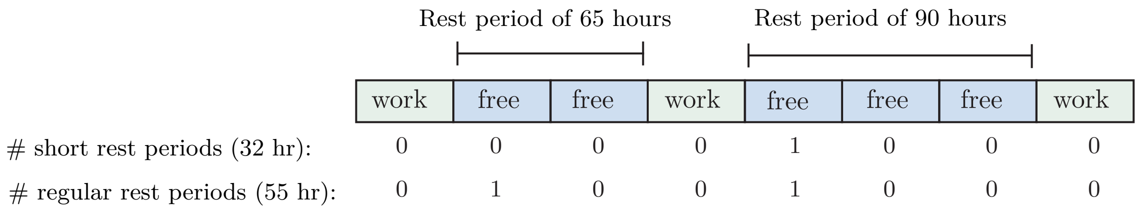

Figure 3.5: Illustration of how short and regular rest periods are counted. For each rest period, the number of regular rest periods assigned is maximized. Afterwards, a short rest period is assigned if the remaining time in the rest period allows it. Both the number of short and regular rest periods are always assigned to the first day in a rest period.

Then, the generic startup constraints can be formulated as:

$$fDH^{\text{start}}_{p,d} \geq fD_{p,d} - fD_{p,d-1}, \qquad \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^{e}] \tag{C2.44}$$

$$fDH^{\text{start}}_{p,d} \leq fD_{p,d}, \qquad \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^{e}] \tag{C2.45}$$

$$fDH^{\text{start}}_{p,d} \leq 1 - fD_{p,d-1}, \qquad \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^{e}] \tag{C2.46}$$

The above generic startup constraints are equivalent to (2), (3), and (4) in Queyranne and Wolsey [2017] and will not be explained more in detail here.

The next step is to define which rest periods should be counted in the current month. The helper variable $fDH^{\text{start}}_{p,d}$ is defined for all rest periods in both the current and previous month, and naturally, it must be ensured that no rest period is counted in more than one roster. The complete duration of a rest period cannot be determined before the subsequent first work shift is assigned. Therefore, any rest period that was not followed by a work shift in the previous month, *is* counted in the current month. Similarly but opposite, any rest period that is not followed by a work shift in the current month, is *not* counted in the current month, but will instead be counted in the next month. For clarity these concept are illustrated by examples in Figure 3.6.
To decide which rest periods to count, a new decision variable $fD^{\text{start}}_{p,d}$ is introduced. $fD^{\text{start}}_{p,d}$ is a proper subset of the helper variable $fDH^{\text{start}}_{p,d}$, and is only one on the first day of a rest period that should be counted in the current month, otherwise zero.
Since it has been assumed that no rest period is longer than 14 days, all rest periods starting in the current month on all days except the last 14 days, $d^{e-13} - d^{e}$, must be counted:

$$fDH^{\text{start}}_{p,d} = fD^{\text{start}}_{p,d}, \qquad \forall p \in \mathcal{P}, d \in [d^{1}, ..., d^{e-14}] \tag{C2.47}$$

Using the same reasoning, all rest periods starting in the previous month on all days except the last 14 days must *not* be counted:

$$fD^{\text{start}}_{p,d} = 0, \qquad \forall p \in \mathcal{P},\ d \in [d^{s}, ..., d^{-15}] \tag{C2.48}$$

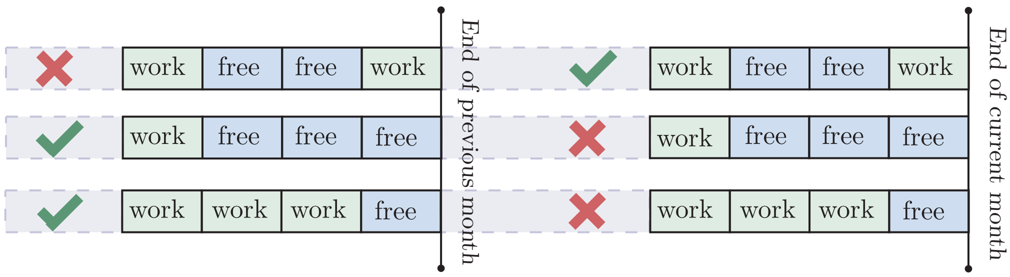

Figure 3.6: Illustration by examples of which rest periods are counted in the current month. Any rest period at the end of the previous month which was not followed by at least one working shift, *is* counted in the current month. Oppositely, any rest period at the end of the current month which is not followed by at least one working shift, is *not* counted in the current month.

3.9.1 Counting rest periods from the end of the previous month

As pictured in Figure 3.6, it must be identified using $fD^{\text{start}}_{p,d}$, which rest periods to count and which not to count in the last two weeks of the previous month. Firstly, consider the upper bound on $fD^{\text{start}}_{p,d}$:

$$fD^{\text{start}}_{p,d} \leq fDH^{\text{start}}_{p,d}, \qquad \forall p \in \mathcal{P},\ d \in [d^{-14}, ..., d^{-1}] \tag{C2.49}$$

The above constraint forces $fD^{\text{start}}_{p,d}$ to be zero on all days where the helper variable $fDH^{\text{start}}_{p,d}$ is zero.

Secondly, consider the lower bound on $fD^{\text{start}}_{p,d}$:

$$fD^{\text{start}}_{p,d} \geq fDH^{\text{start}}_{p,d} + \sum_{d_1=d}^{d^{-1}} fD_{p,d_1} - (d^{-1} - d + 1), \qquad \forall p \in \mathcal{P},\ d \in [d^{-14}, ..., d^{-1}] \tag{C2.50}$$

Here, $fD^{\text{start}}_{p,d}$ will only be forced to one on days d where both of the following statements are true:

1. Day d is the first day in a rest period, i.e. $fDH^{\text{start}}_{p,d} = 1$.
2. Every day from day d up until the last day of the previous month, day d^{-1}, is part of a rest period, i.e. $fD_{p,d} = 1$ for all of these $(d^{-1} - d + 1)$ number of days. Thus, $\sum_{d_1=d}^{d^{-1}} fD_{p,d_1} - (d^{-1} - d + 1)$ becomes exactly zero and otherwise a negative value in all other cases.

In the cases where the above statements are not both true, $fD^{\text{start}}_{p,d}$ is not forced to be either zero nor one.

Lastly, a second upper bound is introduced. This upper bound forces $fD^{\text{start}}_{p,d}$ to be zero on the first day in a rest period, that should *not* be counted:

$$fD^{\text{start}}_{p,d} \leq \frac{\sum_{d_1=d}^{d^{-1}} fD_{p,d_1}}{(d^{-1} - d + 1)}, \qquad \forall p \in \mathcal{P}, d \in \{d^{-14}, ..., d^{-1}\} \tag{C2.51}$$

Here, the right-hand-side can only be equal to one when exactly the second condition from the lower bound (C2.50) is true; i.e. when all the days from day d to the end of the previous month are all part of a rest period. The right-hand-side fraction will be < 1 in all other cases, forcing $fD^{\text{start}}_{p,d}$ to zero. Note that this means that on the second, third, etc. day in a rest period to be counted, $fD^{\text{start}}_{p,d}$ is not bounded to be zero given the above upper bound. Hence, why the other upper bound (C2.49) is also needed.

3.9.2 Counting rest periods at the end of the current month

As pictured in Figure 3.6, it must be identified using $fD^{\text{start}}_{p,d}$, which rest periods to count and which not to count in the last two weeks of the current month. Firstly, consider the upper bound on $fD^{\text{start}}_{p,d}$ equivalent to (C2.49):

$$fD^{\text{start}}_{p,d} \leq fDH^{\text{start}}_{p,d}, \qquad \forall p \in \mathcal{P},\ d \in \{D^{e-13}, ..., d^{e}\} \tag{C2.52}$$

Secondly, consider the second upper bound on $fD^{\text{start}}_{p,d}$, which is similar to (C2.50):

$$\left(1 - fD^{\text{start}}_{p,d}\right) \geq fDH^{\text{start}}_{p,d} + \sum_{d_1=d}^{d^{e}} fD_{p,d_1} - (d^{e} - d + 1) \qquad \forall p \in \mathcal{P},\ d \in [D^{e-13}, ..., d^{e}] \tag{C2.53}$$

Before, (C2.50) was a lower bound forcing $fD^{\text{start}}_{p,d}$ to be one when two conditions were both true. For the same case, the above constraint now acts as an upper bound forcing $fD^{\text{start}}_{p,d}$ to be zero. As shown in Figure 3.6, all rest period cases counted at the end of the previous month, should oppositely *not* be counted at the end of the current month.

Lastly, a lower bound similar to (C2.51) is introduced. The lower bound forces $fD^{\text{start}}_{p,d}$ to be one on the first day in a rest period, that should be counted:

$$fD^{\text{start}}_{p,d} \geq fDH^{\text{start}}_{p,d} - \frac{\sum_{d_1=d}^{d^e} fD_{p,d_1}}{(d^e - d + 1)}, \qquad \forall p \in \mathcal{P},\ d \in [D^{e-13}, ..., d^e] \tag{C2.54}$$

Here, $fD^{\text{start}}_{p,d}$ is forced to be one when the following conditions both are true:

1. $fDH^{\text{start}}_{p,d} = 1$, i.e. day d is the first day in a rest period and then $\sum_{d_1=d}^{d^e} fD_{p,d_1} \geq 1$, i.e. at least one of the remaining days at the end of the month is also part of a rest period, namely day d itself.
2. $\sum_{d_1=d}^{d^e} fD_{p,d_1} < (d^e - d + 1)$, i.e. at least one of the remaining days in the current month is a work day. Thus, the fraction on the right-hand-side will be both < 1 and > 0.

In all other cases the entire right hand side will be ≤ 0.

3.9.3 Determining the number of short and regular rest periods

On the days d that are the first day in a rest period that should be counted, i.e. where $fD^{\text{start}}_{p,d} = 1$, it must be determined how many short and regular rest periods can be assigned. Here, $fD^{\text{couS}}_{p,d}$ is the whole number of short rest periods assigned to day d, and $fD^{\text{couR}}_{p,d}$ is the whole number of regular rest periods assigned to day d.
On all days d where $fD^{\text{start}}_{p,d} = 0$ no short or regular rest periods must be assigned. This is ensure by the following upper bound:

$$45\, fD^{\text{start}}_{p,d} \geq fD^{\text{couS}}_{p,d} + fD^{\text{couR}}_{p,d}, \qquad \forall p \in \mathcal{P},\ d \in [d^s, ..., d^e] \tag{C2.55}$$

Here, $2\text{months} \cdot 30\text{days} \cdot 24\text{hours}/32\text{hours} = 45$ is a sufficiently larger number, as there can never be 45 or more short (32 hours) or regular (55 hours) rest periods in a schedule.

Secondly, it must be ensured that the amount of hours equivalent to the number of short and regular rest periods assigned, must not exceed the time to the next working shift $t^{\text{shiF}}_{p,d}$, i.e. the duration of the rest period:

$$\begin{gathered} t^{\text{shiF}}_{p,d-1} \geq H^{\text{fDS}}_p\, fD^{\text{couS}}_{p,d} + H^{\text{fDR}}_p\, fD^{\text{couR}}_{p,d} - 45\left(1 - fD^{\text{start}}_{p,d}\right), \\ \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^e] \end{gathered} \tag{C2.56}$$

Here, the constraint should only be active on the first day d of a rest period that must be counted, hence when $fD^{\text{start}}_{p,d} = 1$. This is ensured by the last part of the right-hand-side.

When possible, it is preferred to assign a regular rest period over a short rest period. Thus, the number of regular rest periods assigned is maximized with respect to the duration of the rest period:

$$\begin{gathered} \frac{t^{\text{shiF}}_{p,d-1}}{H^{\text{fDR}}_p} - 1 < fD^{\text{couR}}_{p,d} + 45(1 - fD^{\text{start}}_{p,d}) \\ \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^e] \end{gathered} \tag{C2.57}$$

Again, the last part of the right-hand-side turns off the constraint for days on which $fD_{p,d}^{\text{start}} = 0$. The $' - 1'$ on the left hand side is needed when the time to the next working shift divided by the duration of a regular rest period is a fractional number, e.g. 2.5. Then, $fD_{p,d}^{\text{couR}}$ will correctly be lower bounded to 2. When the time to the next working shift divided by the duration of a regular rest period is a whole number, e.g. 2, then $fD_{p,d}^{\text{couR}}$ will still correctly be lower bounded to 2 as well.

After assigning the maximum number of regular rest periods, a short rest period should be given whenever the total remaining time allows it. The following lower bound maximizes the number of short rest periods assigned:

$$\frac{t_{p,d-1}^{\text{shiF}}}{H_p^{\text{fDS}}} - 1 < fD_{p,d}^{\text{couS}} + \frac{H_p^{\text{fDR}}\, fD_{p,d}^{\text{couR}}}{H_p^{\text{fDS}}} + 45\left(1 - fD_{p,d}^{\text{start}}\right), \qquad \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^{e}] \tag{C2.58}$$

Here, the left-hand-side of the constraint calculates how many short rest periods the duration of the entire rest period is equivalent to. On the right-hand-side the number of short rest periods assigned, $fD_{p,d}^{\text{couS}}$, is added to a fraction which converts the number of regular rest periods assigned, $fD_{p,d}^{\text{couR}}$, into the equivalent number of short rest periods. Again, the last part of the right-hand-side turns off the constraint for days on which $fD_{p,d}^{\text{start}} = 0$.

Lastly, it should be ensured that the targeted minimum number of rest periods, FD_p^{min}, for the current schedule is met. Here, the target is typically around four rest periods per month, which ensures that the target amount of rest periods for an entire norm period is met. To avoid infeasibility a slack variable, pFD_p^{u}, is added to capture the number of free periods missing to reach the target, if any. This slack variable is punished in the objective:

$$pFD_p^{\text{u}} \geq FD_p^{\text{min}} - \sum_{d=d^s}^{d^e} \left(\frac{fD_{p,d}^{\text{couS}}}{2} + fD_{p,d}^{\text{couR}} \right), \qquad \forall p \in \mathcal{P} \tag{C2.59}$$

Recall that a short rest period counts for half a rest period, whereas a regular rest period counts for a full rest period. If the combined number of short and regular rest periods exceeds the target, then the punished slack variable can be set to zero.

3.10 Weekend work

According to the Danish working time directive, a physician should not work more than every other weekend on average [Yngre Læger, 2017]. Here, exceptions to the rule are allowed in extraordinary cases. A physician is considered to work on a given weekend if she or he works either Saturday, Sunday, or both days.

As the number of weekends in a month can vary from 4-5, it is not straight forward to ensure that at most half of all weekends in a norm period are working weekends. Using historic data for the current norm period for each employee is also fairly complex, as the duration and beginning date of a norm period varies. Instead, the approach taken is to add a punishment to the objective whenever a physician works two consecutive weekends. If a physician has wished to work two consecutive weekends, then in this case, the objective will not be punished.

First, the binary decision variable $wkd_{p,q}^{\text{work}}$ is introduced for a set of weekends $q \in \mathcal{Q}$. Here, all weekends in the current month, half and whole, are included in $\mathcal{Q}$ as well as the last whole weekend in the previous month. $wkd_{p,q}^{\text{work}}$ is one if physician p has to work on weekend q, otherwise zero. The following constraint is a lower bound for $wkd^{\text{work}}_{p,q}$:

$$\begin{aligned}&\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} S^{\text{work}}_s\left(z_{p,d,s,a}+z_{p,d+1,s,a}\right)\leq 2\,wkd^{\text{work}}_{p,q}\\&\qquad\forall p\in\mathcal{P},\ d\in[d^{-8},...,d^e],\ q\in\mathcal{Q};\ D^{\text{numL}}_d=6\end{aligned}\tag{C2.60}$$

When $D^{\text{numL}}_d = 6$ it means that day d is the 6th day in the week, i.e. Saturday. Thus, $wkd^{\text{work}}_{p,q}$ is forced to be one if a work shift has been assigned to either Saturday or Sunday.

Similarly, the following constraint is a lower bound for $wkd^{\text{work}}_{p,q}$:

$$\begin{aligned}&\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} S^{\text{work}}_s\left(z_{p,d,s,a}+z_{p,d+1,s,a}\right)\geq wkd^{\text{work}}_{p,q}\\&\qquad\forall p\in\mathcal{P},\ d\in[d^{-8},...,d^e],\ q\in\mathcal{Q};\ D^{\text{numL}}_d=6\end{aligned}\tag{C2.61}$$

Based on the shift assignment in the last whole weekend of the previous month (denoted by $q = \{\text{first}\}$), the binary punish variable $pWkd^{\text{work}}_{p,q}$ is forced to one, if the shift assignment on the subsequent weekend results in physician p working on both weekends:

$$\begin{aligned}&wkd^{\text{work}}_{p,q}+wkd^{\text{work}}_{p,q+1}\leq pWkd^{\text{work}}_{p,q+1}+1+W^{\text{wkd}}_{p,q+1}\\&\qquad\forall p\in\mathcal{P},\ q=\{first\}\end{aligned}\tag{C2.62}$$

If a physician has wished to work in the first weekend of the current month denoted by $W^{\text{wkd}}_{p,2} = 1$, knowing his or her schedule for the previous month, then the punish variable will never be activated.

The constraint for the remaining weekends of the month is similar, except now the shift wishes for all subsequent pairwise weekends must be taken into account:

$$\begin{aligned}&wkd^{\text{work}}_{p,q}+wkd^{\text{work}}_{p,q+1}\leq pWkd^{\text{work}}_{p,q+1}+1+\frac{W^{\text{wkd}}_{p,q}+W^{\text{wkd}}_{p,q+1}}{2}\\&\qquad\forall p\in\mathcal{P},\ q\in\mathcal{Q}\backslash\{\text{first}\}\wedge\{\text{last}\}\end{aligned}\tag{C2.63}$$

3.11 Avoiding successive shifts and add-on cases

Certain shifts and add-ons are considered inconvenient to be assigned on two successive days. These shifts and add-ons are marked by a one in column G3 in *Data sheet 2: Shift types*. As an example, it is not preferred to be assigned two successive long shifts.
The following four constraints will turn on a binary punish variable, $p^{\text{row}}_{p,d}$, if one or more of the following cases occur, respectively:

1. Two shifts, where $S^{\text{row}}_s = 1$ for both, are assigned in a row.
2. Two add-ons, where $A^{\text{row}}_a = 1$ for both, are assigned in a row.
3. A shifts, where $S^{\text{row}}_s = 1$, is assigned, directly followed by an add-on, where $A^{\text{row}}_a = 1$.
4. An add-on, where $A^{\text{row}}_a = 1$, is assigned, directly followed by a shift, where $S^{\text{row}}_s = 1$.

$$1+p^{\text{row}}_{p,d}\geq\sum_{s\in\mathcal{S}}\sum_{s\in\mathcal{A}} S^{\text{row}}_s(z_{p,d-1,s,a}+z_{p,d,s,a}),\qquad\forall p\in\mathcal{P},\ d\in[d^1,...d^e]\tag{C2.64}$$

$$1+p^{\text{row}}_{p,d}\geq\sum_{s\in\mathcal{S}}\sum_{s\in\mathcal{A}} A^{\text{row}}_a(z_{p,d-1,s,a}+z_{p,d,s,a}),\qquad\forall p\in\mathcal{P},\ d\in[d^1,...d^e]\tag{C2.65}$$

$$1+p^{\text{row}}_{p,d}\geq\sum_{s\in\mathcal{S}}\sum_{s\in\mathcal{A}} S^{\text{row}}_s\ z_{p,d-1,s,a}+A^{\text{row}}_a\ z_{p,d,s,a},\qquad\forall p\in\mathcal{P},\ d\in[d^1,...d^e]\tag{C2.66}$$

$$1+p^{\text{row}}_{p,d}\geq\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} A^{\text{row}}_a\ z_{p,d-1,s,a}+S^{\text{row}}_s\ z_{p,d,s,a},\qquad\forall p\in\mathcal{P},\ d\in[d^1,...d^e]\tag{C2.67}$$

3.12 Constrained shifts assignments

The assignment of certain shifts or groups of shifts are further constrained than others. These additional constraints are handled here.
First, consider the sleep shift 'SOV'. 'SOV' can be assigned on any day of the week as it must be possible to assign 'SOV' after any night shift. However, it should be not possible to assign 'SOV' in any other case than when a night shift or add-on was assigned on the previous day:

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} z_{p,d,s,a}\left(K^{\text{shi}}_{s,k}+K^{\text{add}}_{a,k}\right) \geq \sum_{a_1\in\mathcal{A}} z_{p,d+1,s_1,a_1}, \qquad \forall p\in\mathcal{P},\ d\in[d^{-1},...,d^{e-1}],\ k=\{Night\},\ s_1=\{SOV\} \tag{C2.68}$$

Here, if a shift or add-on requires a night competency, $k = \{Night\}$, then the shift or add-on occurs during the night.

Certain shifts should not be assigned unless a physician specifically wishes for it, denoted by $S^{\text{ifWi}}_s = 1$. This includes obvious 'off' shifts such as vacations, compensation days, etc., but also includes some work shift such as 'HEMS', i.e. manning the medical helicopter. Here, the schedule for the medical helicopter is the responsibility of a different department, and the physicians within the anesthesia department with the 'HEMS' competency, will know in advance the exact dates on which they must man a 'HEMS' shift. These types of shifts are all marked in column G4 in *Data sheet 2: Shift types*.
Firstly, on all days with no shift wish, i.e. when $WP^{\text{shi}}_{p,d} = \emptyset$, logically a shift that must be wished for cannot be assigned:

$$\sum_{a\in\mathcal{A}} x_{p,d,s,a} = 0, \qquad \forall p\in\mathcal{P},\ d\in\mathcal{D},\ s\in\mathcal{S};\ S^{\text{ifWi}}_s = 1 \wedge WP^{\text{shi}}_{p,d} = \emptyset \tag{C2.69}$$

Secondly, on days with a shift wish, a shift that must be wished for can only be assigned if both the shift and the add-on assigned are allowed given the shift wish, $WP^{\text{shi}}_{p,d}$. E.g. for the shift wish w ='HEMS', the shift assignment s ='HEMS' will be allowed, i.e. $WS^{\text{shi}}_{w,s}$ will be equal to one:

$$\sum_{a\in\mathcal{A}} x_{p,d,s,a} \leq \sum_{a\in\mathcal{A}} WS^{\text{add}}_{w,a}\, WS^{\text{shi}}_{w,s} \qquad \forall p\in\mathcal{P},\ d\in\mathcal{D},\ s\in\mathcal{S},\ w=WP^{\text{shi}}_{p,d};\ S^{\text{ifWi}}_s = 1 \wedge WP^{\text{shi}}_{p,d} \neq \emptyset \tag{C2.70}$$

3.13 Sunday work

The union Yngre Læger (YL) which represents all physicians except chief physicians, has enforced that whenever a physician works on a Sunday, $D^{\text{numL}}_d = 7$, then she or he must be assigned a 'free' day, FRI_h, on one of the following week days; Monday-Friday. If a physician works on a Sunday followed directly by a vacation period, the physician is still entitled to the free day after being back from vacation. However, this special case is not included in this model. To avoid infeasibility, a slack variable, $p^{\text{YL}}_{p,d}$, is activated and punished in the objective every time the rule is violated:

$$\sum_{d_1=d+1}^{d+5;\, d_1\leq d^e}\ \sum_{a\in\mathcal{A}} z_{p,d_1,s_1,a} + p^{\text{YL}}_{p,d} \geq \sum_{s\in\mathcal{S}}\sum_{a_1\in\mathcal{A}} S^{\text{work}}_s\, z_{p,d,s,a_1}, \qquad \forall p\in\mathcal{P},\ d\in[d^{-3},...,d^{e-2}], \quad s_1=\{\text{FRI\_h}\},\ k=\{YL\};\ D^{\text{numL}}_d = 7 \wedge K^{\text{pers}}_{p,k} = 1 \tag{C2.71}$$

Here, if the rule is violated, the slack variable will always be activated on the Sunday on which the YL physician worked. Furthermore, if a Sunday falls on one of the last three days of the month, then there are a most two days left in the current month on which the free day can be assigned. In this case, the free day is assigned in the following month.

To ease the workload of the solver, the slack variable is fixed to zero on all other days than the Sundays considered above:

$$\begin{aligned} & p^{\mathrm{YL}}_{p,d} = 0 \\ & \qquad\qquad \forall p \in \mathcal{P},\ d \in [d^{s}, ..., d^{e}];\ D^{\mathrm{numL}}_{d} \neq 7 \lor d < d^{-3} \lor d > d^{e-2} \end{aligned} \tag{C2.72}$$

3.14 Fraction of chief physicians assigned to presence shifts and add-ons

From a financial standpoint, the anesthesia department should not assign more than $R^{\mathrm{olg}} = 2/3$ of all presence shifts and add-ons to chief physicians (OLG). Recall, a presence shifts or add-ons is a shift at the clinic during the night and is denoted by $RA^{\mathrm{pres}}_{a} = 1$ and $RS^{\mathrm{pres}}_{s} = 1$, respectively. If the R^{olg} threshold is exceeded, a slack variable, p^{OLG}, is activated which will be equal to the fraction which the threshold is exceeded by:

$$\begin{aligned} & \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,d,s,a}\ (RA^{\mathrm{pres}}_{a} + RS^{\mathrm{pres}}_{s})\ K_{p,k} - p^{\mathrm{OLG}} \\ & \quad \leq R^{\mathrm{olg}} \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,d,s,a}\ (RA^{\mathrm{pres}}_{a} + RS^{\mathrm{pres}}_{s}), \qquad k = \{OLG\} \end{aligned} \tag{C2.73}$$

Here, the right-hand-side is the combined number of all presence shift and add-ons in the current month and the left-hand-side is the combined number of all presence shift and add-ons assigned to a chief physician, minus the slack variable.

4 Fairness model

In the *MVP model*, the model assigns shifts/add-ons depending on the elements that affect the objective function. Whenever a shift or add-on wish is not granted the model punishes the objective. Therefore, the model will try to fulfill as many wishes as possible to the extent where it is optimal with respect to all of the elements in the objective. This means that one physician might be granted 15 out of 16 wishes while another may only be granted one of six. From an individual perspective this way of granting wishes seems unfair.
The same logic applies to the amount of over hours hours assigned and the number of shift/add-on assigned for specific counters, such as night or long shift counters. Again, the objective function will simply try to minimize the total number of over hours while still meeting the demand for all the counters. The objective will therefore currently not take into account the distribution of over hours among the physicians or how the shifts/add-ons within each counter is distributed between them. The lack of fairness on the above mentioned elements results in the MVP model generating schedules that are far from fair from the perspective of the individual physician.

The focus of the Fairness model, which is an extension of the *MVP model*, will therefore be to include fairness elements with respect to the percentage of fulfilled wishes, the amount of over hours, and the number of shifts/add-ons assigned within specific counters. First, the additional added sets, parameters, and decisions variables needed to extend the model will be introduced, followed by the fairness model itself.

Sets:

$\mathcal{C}^{\text{f}}$	Set of all counters, $\mathcal{C}^{\text{f}} \subseteq \mathcal{C}$, which fairness is applied to. This subset of counters is identified in *Data sheet 6: Counters* column *Fair*.
$\mathcal{T}$	Set of all steps/levels for the fairness counters. For this model, there a three fairness levels for both fairness on wishes, fairness on over hours, and fairness on counters.

Parameters:

$W^{\text{nStar}}_{p,d}$	$\mathbb{Z}_2$	1 if a physician $p \in \mathcal{P}$ on day $d \in \mathcal{D}$ wishes for either a shift or an add-on and the wish is not marked by a star, otherwise zero.
W^{nStarT}_{p}	$\mathbb{Z}_2$	1 if a physician $p \in \mathcal{P}$ has wished for at least one wish in the current month that is not marked by a star, otherwise zero. Thus, if $\sum_{d \in \mathcal{D}} W^{\text{nStar}}_{p,d} \geq 1$, then $W^{\text{nStarT}}_{p} = 1$.
W^{step}_{t}	$\mathbb{R}^{0+}$	The value of each step $t \in \mathcal{T}$ for fairness on wishes. Each step size is equivalent to a fraction of fulfilled wishes. The fairness wish steps are chosen as: [0.05, 0.20, 0.35].
OH^{step}_{t}	$\mathbb{R}^{0+}$	The value of each step $t \in \mathcal{T}$ for fairness on over hours. Each step size is is equivalent to a fraction of hours per total norm hours in a month. The fairness over hours steps are chosen as: [0, 10/150, 18/150].

C_t^{step}	$\mathbb{R}^{0+}$	The value of each step $t \in \mathcal{T}$ for fairness on counters. Each step size is a number of shifts/add-on per number of days in a month. The fairness counters steps are chosen as: [1/31, 2/31, 3/31].
N_p^{WorkD}	$\mathbb{Z}^{0+}$	The number of days a physician $p \in \mathcal{P}$ can work in a month. This corresponds to the total number of days in the month subtracted by the number of star-wishes that are equivalent to not working, such as a 'vacation' or 'off' wish.
$C_{p,c}^{\text{emp}}$	$\mathbb{Z}_2$	1 if a physician $p \in \mathcal{P}$ has the competency that the counter $c \in \mathcal{C}$ requires, the competencies that at least one shift and one add-on requires, where both the shift and the add-on are valid with respect to the counter, and where the shift and add-on are also compatible, otherwise 0. The parameter, $C_{p,c}^{\text{emp}}$ is 1 when the following statement is true $1 \leq \sum_{k \in \mathcal{K}}(K_{c,k}^{\text{cou}} K_{p,k}^{\text{pers}}) \sum_{s \in \mathcal{S}}(C_{c,s}^{\text{shi}} PS_{p,s}^{\text{pos}}) \sum_{a \in \mathcal{A}}(C_{c,a}^{\text{add}} PA_{p,a}^{\text{pos}}) \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}}(C_{c,s}^{\text{shi}} C_{c,a}^{\text{add}} SA_{s,a}^{\text{com}})$.

Decision variables:

$pF_{p,t}^{\text{w}}$	$\mathbb{R}^{0+}$	The punish value for a physician $p \in \mathcal{P}$ with respect to each step/level $t \in \mathcal{T}$ for fairness on wishes.
$f_{p,d}^{\text{wGive}}$	$\mathbb{Z}_2$	1 if a wish without a star is fulfilled for a physician $p \in \mathcal{P}$ on day $d \in \mathcal{D}$, otherwise zero.
$f^{\overline{\text{w}}}$	$\mathbb{R}^{0+}$	The average percentage of wishes fulfilled without a star for all of the physicians in the current month who wished at least once, i.e. for whom $W_p^{\text{nStarT}} = 1$.
$pF_{p,t}^{\text{oH}}$	$\mathbb{R}^{0+}$	The punish value for a physician $p \in \mathcal{P}$ with respect to each step/level $t \in \mathcal{T}$ for fairness on over hours.
$f^{\overline{\text{oH}}}$	$\mathbb{R}^{0+}$	The average amount of over hours for all the physicians in the current month
$pF_{p,c,t}^{\text{cou}}$	$\mathbb{R}^{0+}$	The punish value for a physician $p \in \mathcal{P}$ with respect to each step/level $t \in \mathcal{T}$ for fairness on counter $c \in \mathcal{C}^{\text{f}}$.
$f_c^{\overline{\text{cou}}}$	$\mathbb{R}^{0+}$	The average number of shifts and add-ons within counter $c \in \mathcal{C}^{\text{f}}$ assigned to physicians that are able to be take on at least one shifts-add-on-combination valid for the counter, i.e. for whom $C_{p,c}^{\text{emp}} = 1$.

4.1 Objective

The first six lines of the objective function in the Fairness model are the same as in the *MVP model*. The three last lines of the objective function are the new fairness elements: fairness on the fraction of fulfilled wishes, amount of over hours, and the fraction of shifts/add-ons for specific counters, respectively. Each fairness element is punished using three levels. Every level is written out in the objective to more clearly visualize the punishment coefficients in relation to the rest of the objectives.

$$
\begin{aligned}
\text{Min} \quad & \sum_{p\in\mathcal{P}} (St_p^{\text{oHr}}\, h_p^{\text{ov}} + 10\, h_p^{\text{un}}) + \sum_{p\in\mathcal{P}} (30\, St_p^{\text{oHr}}\, NR_p\, h_p^{\text{ov}}) + \\
& \sum_{p\in\mathcal{P}} \sum_{c\in\mathcal{C}^{\text{perc}}} (hC_{p,c}^{\text{ov}} + 5\, hC_{p,c}^{\text{un}}) + \sum_{c\in\mathcal{C}^{\text{prio}}} p_c^{\text{prio}} + 10 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} p_{p,d}^{\text{wish}} + \\
& 10 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} p_{p,d}^{\text{adm}} + 10 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} \sum_{s\in\mathcal{S}} p_{p,d,s}^{\text{admS}} + 10 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} \sum_{a\in\mathcal{A}} p_{p,d,a}^{\text{admA}} + \\
& 50 \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} pWkd_{p,q}^{\text{work}} + 100 \sum_{p\in\mathcal{P}} pFD_p^{\text{u}} + 1000\, p^{\text{OLG}} + \\
& 500 \sum_{c\in\mathcal{C}} \sum_{d\in\mathcal{D}} \left(o_{c,d}^{\text{dem}} + u_{c,d}^{\text{dem}}\right) + 5 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} p_{p,d}^{\text{row}} + \\
& 20 \sum_{p\in\mathcal{P}} \left(p_p^{\text{pres}} + p_p^{\text{alert}} + p_p^{\text{call}}\right) + 10 \sum_{p\in\mathcal{P}} \sum_{d\in D^{\text{all}}} p_{p,d}^{\text{YL}} + \\
& \sum_{p\in\mathcal{P}} \left(50\, pF_{p,1}^{\text{w}} + 200\, pF_{p,2}^{\text{w}} + 800\, pF_{p,3}^{\text{w}}\right) + \\
& \sum_{p\in\mathcal{P}} \left(200\, pF_{p,1}^{\text{oH}} + 800\, pF_{p,2}^{\text{oH}} + 2600\, pF_{p,3}^{\text{oH}}\right) + \\
& \sum_{p\in\mathcal{P}} \sum_{c\in\mathcal{C}^{\text{f}}} \left(1000\, pF_{p,c,1}^{\text{cou}} + 2000\, pF_{p,c,2}^{\text{cou}} + 3000\, pF_{p,c,3}^{\text{cou}}\right)
\end{aligned} \tag{O3}
$$

4.2 Fairness on wishes

Fairness on wishes is going to be introduced by adding an increasing punishment to the objective. The further the fraction of fulfilled wishes for a physician is below the average fraction of fulfilled wishes for all physicians in the anesthesia department, the more the punishment will increase. In order to do so the average fraction of wishes fulfilled without a star must be determined. It has been chosen to exclude star-wishes from the fairness calculations as these are wishes that will always be granted. Star-wishes are typically vacation days or contracted work-days that cannot be moved due to other dependencies. These types of wishes are therefore more like a demand/necessity rather than actual wishes. First, it must be determined how many wishes without a star are granted on each day for each physician using the binary decision variable $f_{p,d}^{\text{wGive}}$. This is done using three constraints that are equivalent to (C1.16) - (C1.18), in which it is ensured that all star-wishes are fulfilled. The only difference is that on the right-hand side of the constraints, a '1' is replaced by $f_{p,d}^{\text{wGive}}$:

$$
\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} x_{p,d,s,a}\, WS_{w,a}^{\text{add}}\, WS_{w,s}^{\text{shi}} = f_{p,d}^{\text{wGive}},
$$
$$
\forall p\in\mathcal{P},\, d\in\mathcal{D},\, w=\{WP_{p,d}^{\text{shi}}\};\, WP_{p,d}^{\text{prio}} \neq \{*\} \wedge WP_{p,d}^{\text{shi}} \neq \emptyset \wedge WP_{p,d}^{\text{add}} = \emptyset \tag{C3.1}
$$

$$
\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} x_{p,d,s,a}\, WA_{w,a}^{\text{add}}\, WA_{w,s}^{\text{shi}} = f_{p,d}^{\text{wGive}},
$$
$$
\forall p\in\mathcal{P},\, d\in\mathcal{D},\, w=\{WP_{p,d}^{\text{add}}\};\, WP_{p,d}^{\text{prio}} \neq \{*\} \wedge WP_{p,d}^{\text{shi}} = \emptyset \wedge WP_{p,d}^{\text{add}} \neq \emptyset \tag{C3.2}
$$

$$
\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} x_{p,d,s,a}\, WS_{w_1,a}^{\text{add}}\, WS_{w_1,s}^{\text{shi}}\, WA_{w_2,a}^{\text{add}}\, WA_{w_2,s}^{\text{shi}} = f_{p,d}^{\text{wGive}}, \qquad \forall p\in\mathcal{P},\, d\in\mathcal{D}, \tag{C3.3}
$$
$$
w_1=\{WP_{p,d}^{\text{shi}}\},\, w_2=\{WP_{p,d}^{\text{add}}\};\, WP_{p,d}^{\text{prio}} \neq \{*\} \wedge WP_{p,d}^{\text{shi}} \neq \emptyset \wedge WP_{p,d}^{\text{add}} \neq \emptyset
$$

Lastly, to ease the workload of the solver, $f_{p,d}^{\text{wGive}}$ is fixed to zero on days with no wishes or a star-wish:

$$f_{p,d}^{\text{wGive}} = 0 \qquad \forall p \in \mathcal{P},\ d \in \mathcal{D};\ W_{p,d}^{\text{nStar}} = 0 \tag{C3.4}$$

Now that $f_{p,d}^{\text{wGive}}$ is defined, the average fraction of fulfilled wishes, $f^{\overline{\text{w}}}$, can be determined:

$$f^{\overline{\text{w}}} = \frac{\sum_{p \in \mathcal{P};\, W_p^{\text{nStarT}}=1} \left(\frac{\sum_{d \in \mathcal{D}} f_{p,d}^{\text{wGive}}}{\sum_{d \in \mathcal{D}} W_{p,d}^{\text{nStar}}} \right)}{\sum_{p \in \mathcal{P}} W_p^{\text{nStarT}}} \tag{C3.5}$$

Here, the numerator first calculates the total number of granted wishes without a star for each physician, over the total number of wishes without a star. This gives the fraction of fulfilled wishes without a star for each physician, except for those physicians who had no wishes without a star, e.g. for whom $W_p^{\text{nStarT}} = 0$ (avoiding division by zero). Then, the numerator sums over all the fractions of fulfilled wishes without a star and finally this number is divides by the number of physicians who wished for at least one wish without a star, i.e. for whom $W_p^{\text{nStarT}} = 1$.

Lastly, the punish value, $pF_{p,t}^{\text{w}}$, for each fairness level t is defined:

$$pF_{p,t}^{\text{w}} \geq f^{\overline{\text{w}}} - \frac{\sum_{d \in \mathcal{D}} f_{p,d}^{\text{wGive}}}{\sum_{d \in \mathcal{D}} W_{p,d}^{\text{nStar}}} - W_t^{\text{step}} \qquad \forall p \in \mathcal{P},\ t \in \mathcal{T};\ W_p^{\text{nStarT}} = 1 \tag{C3.6}$$

As illustrated in Figure 4.1, each fairness level is turned on when the fraction of fulfilled wishes without a star, is below a certain threshold of the average, $f^{\overline{\text{w}}}$. Level 1 is turned on when the fraction is 5% below the average, level 2 is turned on at 20%, and level 3 is turned on at 35%. Thus, in the above constraint the punish value for each level is lower bounded by the average fraction of fulfilled wishes, minus the the fraction of wishes fulfilled for physician p, minus the step size , W_t^{step}, for fairness level t.
As an example, if the average fraction is $f^{\overline{\text{w}}} = 0.8$, and a physician gets 0.5 of her or his

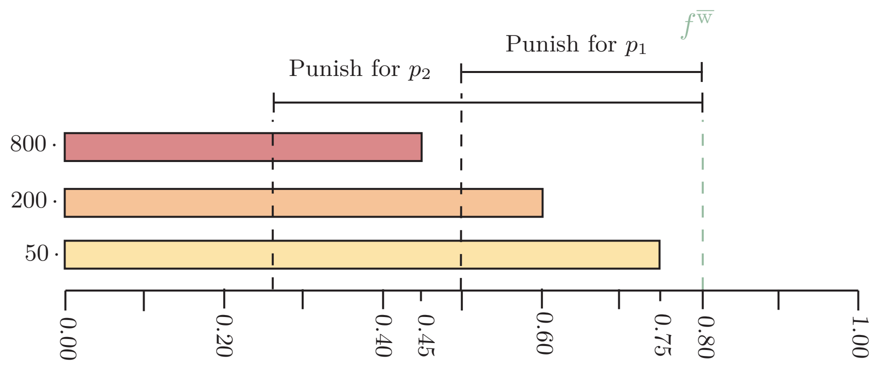

Figure 4.1: Illustration of how the fairness punishment for two physicians, p_1 and p_2, is calculated with respect to the fraction of wishes without a star that has been granted. The yellow bar illustrates the 1st level of punishment, the orange bar the 2nd level, and the red bar the 3rd level. The 'length' of each bar with respect to the fraction of fulfilled wishes for a physician, will be punished in the objective by a factor of 50, 200, and 800, respectively.

wishes fulfilled, then the punishment variable for level 1 becomes: $W_1^{\text{step}} = 0.8 - 0.5 - 0.05 = 0.25$, and the punishment variable for level 2 becomes: $W_2^{\text{step}} = 0.8 - 0.5 - 0.2 = 0.1$. Level 3 will not we activated. Each punishment level is then amplified in the objective by a factor of; 50, 200, and 800, respectively. This example is also pictured in Figure 4.1.

4.3 Fairness on over hours

The same approach as pictured in Figure 4.1, is taken when applying fairness to over hours.

First, the average amount of over hours in relation to the monthly target amount of hours, $f^{\overline{\text{oH}}}$, is calculated:

$$f^{\overline{\text{oH}}} = \frac{\sum_{p \in \mathcal{P};\, NR_p \neq 1} \frac{h_p^{\text{ov}}}{H_p^{\text{targ}}}}{\sum_{p \in \mathcal{P}} (1 - NR_p)} \tag{C3.7}$$

Here, the amount of over hours, h_p^{ov}, for each physician is divided by the hourly target of the month, H_p^{targ}. Summing up over all of these fractions and dividing by the number of physicians, will return the average. Physicians who are not a normal resource, i.e. for whom $NR_p = 1$, are excluded from the calculation, as their monthly target is always zero hours.

Lastly, the punish value, $pF_{p,t}^{\text{oH}}$, for each fairness level t is defined:

$$\begin{aligned} pF_{p,t}^{\text{oH}} \geq \frac{h_p^{\text{ov}}}{H_p^{\text{targ}}} - f^{\overline{\text{oH}}} - OH_t^{\text{step}}, \\ \forall p \in \mathcal{P},\, t \in \mathcal{T};\, NR_p \neq 1 \end{aligned} \tag{C3.8}$$

Similar to how the punish value was decided in (C3.6), the average, $f^{\overline{\text{oH}}}$, and the step size, OH_t^{step} are both subtracted from the fraction which relates the amount of assigned over hours to the total target hours.

4.4 Fairness on counters

The same approach as pictured in Figure 4.1, is also taken when applying fairness to counters. Fairness is only applied to the counters specified by the scheduler in column 'Fair' in *Data sheet 6: Counters.* These include counters such as; long shifts, night shifts, etc.

Firstly, the average fraction of shifts assigned to each fairness counter with respect to the number of possible working days in the month, N_p^{WorkD}, is calculated:

$$f_c^{\overline{\text{cou}}} = \frac{\sum_{p \in \mathcal{P}} \left(\frac{\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} C_{p,c,s,a}^{\text{pos}}\, x_{p,s,a}^{\text{help}}}{N_p^{\text{WorkD}}} \right)}{\sum_{p \in \mathcal{P}} C_{p,c}^{\text{emp}}} \qquad \forall c \in \mathcal{C}^{\text{f}} \tag{C3.9}$$

The idea behind not simply considering the number of shifts assigned within a counter, e.g. four long shift within the long shift counter 't_LONG', is that the availability level of a physician should be reflected in the number. A physician who is on vacation during two weeks of a month should not be assigned as many night shift, long shifts, etc., as a physician who is available to work the entire month. Here, N_p^{WorkD} is the number of days in a month subtracted by vacation and star-marked 'off' wishes, i.e. the availability of a physician.

In order to count the number of shifts assigned to a physician under a counter c, the parameter $C^{\text{pos}}_{p,c,s,a}$ is needed. $C^{\text{pos}}_{p,c,s,a}$ is one if an assigned shift-add-on-combination is counted by counter c and if physician p has the competencies required for both the counter, the shift, and the add-on. Recall from (C1.12) that $x^{\text{help}}_{p,s,a}$ is the summation of each pairwise shift and add-on combination assigned during the entire month. Thus, the numerator in (C3.9) defines exactly the fraction of shifts versus the availability of each physician for each counter. The denominator, the sum of $C^{\text{emp}}_{p,c}$, counts the number of physicians who can be assigned at least one shift-add-on-combination counted by each counter c. Here, it would not be fair to include all physicians in the denominator, as some physicians do not have the prerequisites needed for some counters. Thus, including these physicians would obscure the average, $f^{\overline{\text{cou}}}_c$.

Secondly, the punish value, $pF^{\text{cou}}_{p,c,t}$, for each fairness level t is defined:

$$pF^{\text{cou}}_{p,c,t} \geq \frac{\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C^{\text{pos}}_{p,c,s,a}\, x^{\text{help}}_{p,s,a}}{N^{\text{WorkD}}_p} - f^{\overline{\text{cou}}}_c - C^{\text{step}}_t, \qquad \forall p\in\mathcal{P}, c\in\mathcal{C}^{\text{f}}, t\in\mathcal{T} \tag{C3.10}$$

Here, the approach is identical to how the punish value, $pF^{\text{oH}}_{p,t}$, was defined in (C3.8)

The full fairness model, all constraints included, is enclosed in Appendix A.

5 Performance improvements of the Fairness model

A time analysis of the complete *Fairness model* is presented later in *Optimality analysis of the full Fairness model* as well as an analysis of the practical quality of the solutions returned by the model, which is presented in *Practical quality of the Fairness model*. Here, it is shown that the running time of the complete model is significant, whereas the quality of the solutions at this complexity level is both satisfactory to the scheduler at the anesthesia department, as well as effective in terms of key performance indicators considered. Instead of further increasing the complexity of the complete *Fairness model* by adding more qualitative elements, it was chosen to focus on improving the performance of the model in terms of bounds and time instead.
Two approaches for how to potentially improve the performance of the complete *Fairness model* are presented in this chapter. The first approach uses lexicographic optimization, and the second approach uses decomposition.

5.1 Two stage model

The two stage model uses lexicographic optimization to solve the multi-objective *Fairness model* in two sequential optimization stages. In the first stage, the part of the *Fairness model* which is equivalent to the *MVP model*, is optimized. This includes constraints (C1.1) - (C1.21) introduced in the *Base model* and constraints (C2.1) - (C2.73) introduced in the *MVP model*.

The objective function in the first stage is the objective of the *MVP model*, (O2). For clarity, the objective is restated here.

Stage 1:

$$\begin{aligned}
\text{Min} \quad & \sum_{p\in\mathcal{P}}(St_p^{\text{oHr}}\, h_p^{\text{ov}} + 10\, h_p^{\text{un}}) + \sum_{p\in\mathcal{P}}(30\, St_p^{\text{oHr}}\, NR_p\, h_p^{\text{ov}}) + \\
& \sum_{p\in\mathcal{P}}\sum_{c\in\mathcal{C}^{\text{perc}}}(hC_{p,c}^{\text{ov}} + 5\, hC_{p,c}^{\text{un}}) + \sum_{c\in\mathcal{C}^{\text{prio}}} p_c^{\text{prio}} + 10\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}} p_{p,d}^{\text{wish}} + \\
& 10\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}} p_{p,d}^{\text{adm}} + 10\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}} p_{p,d,s}^{\text{admS}} + 10\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{a\in\mathcal{A}} p_{p,d,a}^{\text{admA}} + \\
& 50\sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{Q}} pWkd_{p,q}^{\text{work}} + 100\sum_{p\in\mathcal{P}} pFD_p^{\text{u}} + 1000\, p^{\text{OLG}} + \\
& 500\sum_{c\in\mathcal{C}}\sum_{d\in\mathcal{D}}\left(o_{c,d}^{\text{dem}} + u_{c,d}^{\text{dem}}\right) + 5\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}} p_{p,d}^{\text{row}} + \\
& 20\sum_{p\in\mathcal{P}}\left(p_p^{\text{pres}} + p_p^{\text{alert}} + p_p^{\text{call}}\right) + 10\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}^{all}} p_{p,d}^{\text{YL}}
\end{aligned} \tag{O4.1}$$

Subject to

(C1.1) - (C1.21)
(C2.1) - (C2.73)

The second stage includes all of the constraints in the complete *Fairness model*; (C1.1) - (C1.21) which were introduced in the *Base model*, (C2.1) - (C2.73) which were introduced

in the *MVP model*, and (C3.1) - (C3.10) which were introduced in the *Fairness model*. Furthermore, a new constraint is added which on the left-hand-side contains the objective function from Stage 1, (O4.1), and on the right-hand-side contains the objective value, $O4.1^{\text{val}}$, from Stage 1 times a certain amount of slack; $Slack^{\text{s1}}$. The value of the fairness decision variables are typically improved at the expense of the optimality of other decision variables, hence why the objective value from Stage 1 is allowed to worsen by a factor of $Slack^{\text{s1}}$. The value of the slack parameter, $Slack^{\text{s1}}$, is chosen to be 15% based on an analysis carried out in *Parameter study of the Two Stage model*.
The values of each decision variable in the best found solution in Stage 1, are used to warm start Stage 2. Thus, every decision variable in Stage 2, except the fairness variables introduced in Table 4, is given an initial starting point equivalent to the solution in Stage 1.

The objective function in the second stage corresponds to the three last lines of the objective function, (O3), from the *Fairness model*. This part of the objective contains exactly all fairness elements exclusively.

Stage 2:

$$
\begin{aligned}
\text{Min} \quad & \sum_{p \in \mathcal{P}} \left(50\, pF^{\text{w}}_{p,1} + 200\, pF^{\text{w}}_{p,2} + 800\, pF^{\text{w}}_{p,3}\right) + \\
& \sum_{p \in \mathcal{P}} \left(200\, pF^{\text{oH}}_{p,1} + 800\, pF^{\text{o}}_{p,2} + 2600\, pF^{\text{oH}}_{p,3}\right) + \\
& \sum_{p \in \mathcal{P}} \sum_{c \in \mathcal{C}^{\text{f}}} \left(1000\, pF^{\text{cou}}_{p,c,1} + 2000\, pF^{\text{cou}}_{p,c,2} + 3000\, pF^{\text{cou}}_{p,c,3}\right)
\end{aligned}
\tag{O4.2}
$$

Subject to

$$
\begin{aligned}
& \sum_{p \in \mathcal{P}} (St^{\text{oHr}}_p\, h^{\text{ov}}_p + 10\, h^{\text{un}}_p) + \sum_{p \in \mathcal{P}} (30\, St^{\text{oHr}}_p\, NR_p\, h^{\text{ov}}_p) + \\
& \sum_{p \in \mathcal{P}} \sum_{c \in \mathcal{C}^{\text{perc}}} (hC^{\text{ov}}_{p,c} + 5\, hC^{\text{un}}_{p,c}) + \sum_{c \in \mathcal{C}^{\text{prio}}} p^{\text{prio}}_c + 10 \sum_{p \in \mathcal{P}} \sum_{d \in \mathcal{D}} p^{\text{wish}}_{p,d} + \\
& 10 \sum_{p \in \mathcal{P}} \sum_{d \in \mathcal{D}} p^{\text{adm}}_{p,d} + 10 \sum_{p \in \mathcal{P}} \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} p^{\text{admS}}_{p,d,s} + 10 \sum_{p \in \mathcal{P}} \sum_{d \in \mathcal{D}} \sum_{a \in \mathcal{A}} p^{\text{admA}}_{p,d,a} + \\
& 50 \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}} pWkd^{\text{work}}_{p,q} + 100 \sum_{p \in \mathcal{P}} pFD^{\text{u}}_p + 1000\, p^{\text{OLG}} + \\
& 500 \sum_{c \in \mathcal{C}} \sum_{d \in \mathcal{D}} \left(o^{\text{dem}}_{c,d} + u^{\text{dem}}_{c,d}\right) + 5 \sum_{p \in \mathcal{P}} \sum_{d \in \mathcal{D}} p^{\text{row}}_{p,d} + \\
& 20 \sum_{p \in \mathcal{P}} \left(p^{\text{pres}}_p + p^{\text{alert}}_p + p^{\text{call}}_p\right) + 10 \sum_{p \in \mathcal{P}} \sum_{d \in D^{\text{all}}} p^{\text{YL}}_{p,d} \leq \left(1 + Slack^{\text{s1}}\right)\, O4.1^{\text{val}}
\end{aligned}
\tag{C4.2.1}
$$

(C1.1) - (C1.21)
(C2.1) - (C2.73)
(C3.1) - (C3.10)

5.2 Decomposition model

For large-scale models which contain decoupled constraints at large, decomposition is a popular approach for both improving solution bounds as well as running time. Within the airline industry, various decomposition methods have been successful in both crew scheduling [Clarke and Ryan, 2013; Lavoie et al., 1988] as well as schedule recovery (typically flight rescheduling) [Lavoie et al., 1988]. In crew scheduling, one decomposition approach is to decompose the problem into separate subproblems for each crew member. Here, the rules concerning the roster of each crew member such as vacation, rest rules, working hours, etc. are uncoupled from the rest of the problem. Column generation can therefore readily be used to generate sets of rosters for each crew member. The set of coupled constraints in the problem, such as fulfilling the daily crew demand for a fleet of air-crafts, are then satisfied in the master problem by choosing one of the generated rosters for each crew member. This classic approach of splitting the constraints into a set of coupling (sometimes referred to as complicating) and uncoupling constraints and subsequently solving each set of constraints in a master- and n subproblems, respectively, is known as Dantzig Wolfe Decomposition [Dantzig and Wolfe, 1960]. Models which can be decomposed using Dantzig Wolfe are said to have block angular structures with respect to the reformulation of the constraint matrix.
This model structure is also seen in the nurse rostering problem within the healthcare industry. Here, various approaches including decomposition- and column generation-based methods have also been explored, as reviewed in Burke et al. [2004] and Cheang et al. [2003].
The block angular structures are also found in scheduling of physicians problems. i.e. the problem at hand. Investigating whether decomposition can improve the running time or the solution bounds therefore seems an obvious approach.

Initially, we choose to decompose the *MVP model* into a set of subproblem; one for each of the physicians in the anesthesia department. Although each subproblem contains the same abstract constraints, the subproblems are not identical due to differences in targets, personal wishes, etc. The master problem contains a lambda reformulation of all the decision variables in the subproblems. Additionally, it contains the two decision variables for the over demand, $o^{\text{dem}}_{c,d}$, and the under demand, $u^{\text{dem}}_{c,d}$, which are the only decision variables which are part of a coupling constraint in the MVP model. Lastly, the two coupling constraints in the master problem for the over and under demand are also reformulated with respect to pattern selection.
It was chosen to test the potential of decomposition on the *MVP model* before decomposing the full *Fairness model*. This is because all of the fairness constraints are coupling constraints which adds significant complexity to the master problem. Thus, if decomposing the MVP model does not add any apparent value to the optimization process of the MVP model, then it is unlikely that decomposing the full Fairness model will be worth the effort within the time-frame of this thesis.

The Decomposition model is presented below. To initiate the the column generation process an initial solution which is feasible with respect to the master problem is needed. Since the two coupling demand constraints in the master problem are implemented as soft constraints using a slack variable, the initial solution provided simply assigns an off shift, 'FRI_h', to each physician on each day of the roster.
To save time, only the objective function of each subproblem is reloaded in each iteration and not all of the constraints.

5.2.1 Master problem

Besides the sets, parameters, and decision variables already introduced in the *MVP model*, a few new additions are needed to construct the master problem.

Sets:

$\mathcal{Y}_p$ The set of patterns for each physician $p \in \mathcal{P}$.

Parameters:

In the master problem all the decision variables used in the subproblems are converted into parameters. To clearly show this difference, the parameter version of each decision variable is renamed as the combination of X__ and the name of the decision variable. As an example, the decision variable $x_{p,d,s,a}$ becomes $X\_x_{p,d,s,a,y}$ in its parameter form. Here, the pattern $y \in \mathcal{Y}_p$ is added as an extra dimensions since each 'new' parameter is the convex combination of extreme points corresponding to the set of solutions generated by each of the $p \in \mathcal{P}$ subproblems. The new parameters originating from the decision variables in the subproblems are:

$$\begin{array}{cccccc} X\_x_{p,d,s,a,y} & X\_h^{\text{ov}}_{p,y} & X\_h^{\text{un}}_{p,y} & X\_hC^{\text{ov}}_{p,c,y} & X\_hC^{\text{un}}_{p,c,y} & X\_p^{\text{prio}}_{p,c,y} \\ X\_p^{\text{admA}}_{p,d,a,y} & X\_p^{\text{admS}}_{p,d,s,y} & X\_p^{\text{adm}}_{p,d,y} & X\_p^{\text{wish}}_{p,d,y} & X\_p^{\text{pres}}_{p,y} & X\_p^{\text{alert}}_{p,y} \\ X\_p^{\text{call}}_{p,y} & X\_pFD^{\text{u}}_{p,y} & X\_pFD^{\text{oU}}_{p,y} & X\_pWkd^{\text{work}}_{p,w,y} & X\_p^{\text{row}}_{p,d,y} & X\_p^{\text{YL}}_{p,d,y} \end{array}$$

Moreover, two new dual parameters are introduced:

$\pi_{c,d}$	$\mathbb{R}^{0+}$	The dual value of the constraint regarding the demand for each counter $c \in \mathcal{C}$ on each day $d \in \mathcal{D}$. This concerns both (C5.1) and (C5.2) as exactly one of these constraints apply to each of the counters $c \in \mathcal{C}$.
κ_p	$\mathbb{R}^{0+}$	The dual value for the convexity constraint for each physician $p \in \mathcal{P}$.

Decision variables:

$\lambda_{p,y}$	$\mathbb{R}^{0+}$	Defines the different patterns/solution $y \in \mathcal{Y}_p$ generated by each subproblem for each physician $p \in \mathcal{P}$.

The master problem couples the solutions, or patterns, generated by each of the subproblems by selecting a convex combination of extreme points.
The objective function is very similar to the objective function (O2) of the MVP model. The main difference is that all parts of the objective are changed to a lambda formulation. Additionally, the part of the objective which concerns the fraction of chief physicians manning nightly add-ons or shifts, p^{OLG}, is taken out to reduce the complexity of the master problem as it corresponds to the coupling constraint (C2.73). This element of the model is not a strict necessity for a valid roster.

$$
\begin{aligned}
\text{Min} \quad & \sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}(St_p^{\text{oHr}}\ X\_h_{p,y}^{\text{ov}}\ \lambda_{p,y}) + 5\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}(X\_h_{p,y}^{\text{un}}\ \lambda_{p,y}) + \\
& 10\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}(St_p^{\text{oHr}}\ NR_p\ X\_h_{p,y}^{\text{ov}}\ \lambda_{p,y}) + \sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}\sum_{c\in\mathcal{C}^{\text{perc}}}(X\_hC_{p,c,y}^{\text{ov}}\ \lambda_{p,y}) + \\
& 5\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}\sum_{c\in\mathcal{C}^{\text{perc}}}(X\_hC_{p,c,y}^{\text{un}}\ \lambda_{p,y}) + \sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}\sum_{c\in\mathcal{C}^{\text{prio}}}(X\_p_{p,c,y}^{\text{prio}}\ \lambda_{p,y}) + \\
& 10\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}(X\_p_{p,d,y}^{\text{wish}}\ \lambda_{p,y}) + 10\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}(X\_p_{p,d,y}^{\text{adm}}\ \lambda_{p,y}) + \\
& 10\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}}(X\_p_{p,d,s,y}^{\text{admS}}\ \lambda_{p,y}) + 10\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{a\in\mathcal{A}}(X\_p_{p,d,a,y}^{\text{admA}}\ \lambda_{p,y}) + \\
& 50\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}\sum_{w\in\mathcal{Q}}(X\_pWkd_{p,w,y}^{\text{work}}\ \lambda_{p,y}) + 100\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}(X\_pFD_{p,y}^{\text{u}}\ \lambda_{p,y}) + \\
& 500\sum_{c\in\mathcal{C}}\sum_{d\in\mathcal{D}}\left(o_{c,d}^{\text{dem}} + u_{c,d}^{\text{dem}}\right) + 5\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}(X\_p_{p,d,y}^{\text{row}}\ \lambda_{p,y}) + \\
& 20\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}\left((X\_p_{p,y}^{\text{pres}} + X\_p_{p,y}^{\text{alert}} + X\_p_{p,y}^{\text{call}})\ \lambda_{p,y}\right) + \\
& 10\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}^{all}}(X\_p_{p,d,y}^{\text{YL}}\ \lambda_{p,y})
\end{aligned}
\tag{O5}
$$

The coupling constraints (C1.10) and (C1.11) regarding the over and under demand on counters are added to the master problem. However, the left-hand-side of each constraint is changed to a lambda reformulation:

Subject to

$$
\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C_{p,c,s,a}^{\text{pos}}\ X\_x_{p,d,s,a,y}\ \lambda_{p,y} = C_{c,d}^{\text{dem}} + o_{c,d}^{\text{dem}} - u_{c,d}^{\text{dem}} \qquad \forall d\in\mathcal{D}, c\in\mathcal{C};\ C_c^{\text{type}} = \{P\} \tag{C5.1}
$$

$$
\sum_{y\in\mathcal{Y}_p}\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C_{p,c,s,a}^{\text{pos}}\ X\_x_{p,d,s,a,y}\ \lambda_{p,y} \geq C_{c,d}^{\text{dem}} - u_{c,d}^{\text{dem}} \qquad \forall d\in\mathcal{D}, c\in\mathcal{C};\ C_c^{\text{type}} = \{M\} \tag{C5.2}
$$

Lastly, the standard formulation of the convexity constraint is added:

$$
\sum_{y\in\mathcal{Y}_p}\lambda_{p,y} = 1 \qquad \forall p\in\mathcal{P} \tag{C5.3}
$$

5.2.2 Subproblems

The purpose of each subproblem is to continue to generate solution patterns for each physician as long as the objective value of the subproblem is negative. The objective value of each subproblem represents the reduced cost of a found solution. Thus, when this value is negative, it means that the found solution/pattern can potentially improve the solution of the master problem.

Besides the sets, parameters, and decision variables already introduced in the *MVP model*, a few new additions are needed to construct the subproblems.

Parameters:

C_c^{typeP}	$\mathbb{Z}_2$	1 if counter $c \in \mathcal{C}$ is a precise counter, otherwise 0. This parameter is 1 when $C_c^{\text{type}} =$ "P".
C_c^{typeM}	$\mathbb{Z}_2$	1 if counter $c \in \mathcal{C}$ is a minimum counter, otherwise 0. This parameter is 1 when $C_c^{\text{type}} =$ "M".

Decision variables:

$x_{d,s,a}^{\text{sub}}$	$\mathbb{Z}_2$	1 if the physician which the subproblem represents is assigned shift $s \in \mathcal{S}$ in combinations with add-on $a \in \mathcal{A}$ on day $d \in \mathcal{D}$, otherwise 0.

The objective function of each subproblem contains all the elements from the objective function, (O2), of the MVP model except the demand objectives. Because a subproblem is created for each of the physicians, the objective function now only considers each objective with respect to a single physician. Additionally, the dual value from the convexity constraint, κ_p, and the demand constraints, $\pi_{c,d}$, are added:

$$\begin{aligned}
\text{Min} \quad & (St_p^{\text{oHr}}\, h_p^{\text{ov}} + 5\, h_p^{\text{un}}) + (10\, St_p^{\text{oHr}}\, NR_p\, h_p^{\text{ov}}) + \\
& \sum_{c \in \mathcal{C}^{\text{perc}}} (hC_{p,c}^{\text{ov}} + 5\, hC_{p,c}^{\text{un}}) + \sum_{c \in \mathcal{C}^{\text{prio}}} p_c^{\text{prio}} + 10 \sum_{d \in \mathcal{D}} p_{p,d}^{\text{wish}} + \\
& 10 \sum_{d \in \mathcal{D}} p_{p,d}^{\text{adm}} + 10 \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} p_{p,d,s}^{\text{admS}} + 10 \sum_{d \in \mathcal{D}} \sum_{a \in \mathcal{A}} p_{p,d,a}^{\text{admA}} + \\
& 50 \sum_{w \in \mathcal{Q}} pWkd_{p,w}^{\text{work}} + 100\, pFD_p^{\text{u}} + 5 \sum_{d \in \mathcal{D}} p_{p,d}^{\text{row}} + \\
& 20 \left(p_p^{\text{pres}} + p_p^{\text{alert}} + p_p^{\text{call}} \right) + 10 \sum_{d \in \mathcal{D}^{all}} p_{p,d}^{\text{YL}} - \\
& \sum_{c \in \mathcal{C}} \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left(\pi_{c,d} \, (C_c^{\text{typeP}} + C_c^{\text{typeM}})\, C_{p,c,s,a}^{\text{pos}}\, x_{d,s,a}^{\text{sub}} \right) - \\
& \kappa_p
\end{aligned} \tag{O6}$$

The constraints included in the MVP model, (C1.1)-(C1.21) and (C2.1)-(C2.73), are equivalent to the constraints in the subproblem except (C1.10) and (C1.11) which are omitted because they specify the over and under demand on counters. Moreover, the sum over all physicians is removed in constraint (C1.19) - (C1.21). Conclusively, all the constraints in each subproblem are:

Subject to

(C1.1) - (C1.9)

(C1.12) - (C1.18)

$$\sum_{s \in \mathcal{S}} \sum_{k \in \mathcal{K}} \sum_{a \in \mathcal{A}} C_{p,c,k,s,a}^{\text{pos}}\, x_{p,s,a}^{\text{help}}\, PP_{c,s}^{\text{shi}} \le p_c^{\text{prio}}, \qquad \forall c \in \mathcal{C}^{\text{prio}};\ C_c^{\text{prio}} = \{\text{Shift}\} \tag{C6.1}$$

$$\sum_{s\in\mathcal{S}}\sum_{k\in\mathcal{K}}\sum_{a\in\mathcal{A}} C^{\text{pos}}_{p,c,k,s,a}\ x^{\text{help}}_{p,s,a}\ PP^{\text{add}}_{c,a} \leq p^{\text{prio}}_{c}, \qquad \forall c \in \mathcal{C}^{\text{prio}};\ C^{\text{prio}}_{c} = \{\text{Add-on}\} \tag{C6.2}$$

$$\sum_{s\in\mathcal{S}}\sum_{k\in\mathcal{K}}\sum_{a\in\mathcal{A}} C^{\text{pos}}_{p,c,k,s,a}\ x^{\text{help}}_{p,s,a}\ \left(1 - \sum_{k1\in\mathcal{K}} K^{\text{emp}}_{p,k1}\ BP^{\text{comp}}_{c,k}\right) \leq p^{\text{prio}}_{c}, \quad \forall c \in \mathcal{C}^{\text{prio}};\ C^{\text{prio}}_{c} = \{\text{Competency}\} \tag{C6.3}$$

(C2.1) - (C2.73)

6 Analysis and results

This chapter details the different analyses performed on the MVP, Fairness, Two stage and decomposition model. Ideally, the various analyses would be carried out for a variety of rosters, but for the purpose of this thesis we have chosen to limit the scope and only consider the schedules of July and September. These two months were specifically picked as July is a holiday month with reduced activity levels and September is a regular month. Hence, the two months are different from one another. July is also the hardest month to schedule, according to the scheduler, due to the holidays. Another difference between July and September is that the schedule of July has 230 un-starred wishes whereas the schedule of September has 76. This is due to scheduler choosing to star the majority of the working wishes in September, thus ensuring working wishes are granted.
The *Fairness model* was used to generate the actual schedule used in the anesthesia department for both the month of July and September. Thus, based on the feedback given from the scheduler, the practical quality of these two instances is known to be both satisfactory and comparable.
The *Base model* will not be part of the analysis since is too basic to provide schedules that are compliant with the Danish Working Time Directive.

All models have been implemented using the Julia programming language[1] and is solved using the commercial solver Gurobi 9.1.1[2]. All experiments have been performed with the default seed (zero) in Gurobi to limit solution variations associated with the randomness in the solver. Furthermore, all analyses have been run on the DTU management cluster on 32 16-core nodes (Xeon Gold 6226R, 150W, 2.90GHz).

6.1 Optimality analysis of the full Fairness model

The purpose of this analysis is to examine the performance of the Fairness model with respect to theoretical optimality. The problem size of each of the two instances before and after preprocessing, also known as presolve in Gurobi, is presented in Table 6.1. Preprocessing is an important aspect of real-world implementations as it can remove redundant information and strengthen variable bounds [Lodi, 2013].
The full Fairness model consists of 104 constraints of which 101 are abstract. This corresponds to roughly 900,000 real constraints before presolve and 45,000 after presolve for both July and September. The total solution space of the model, rows (constraints) multiplied with columns (variables), is $\sim 2e^{12}$, however non-zeros account for 0.1‰ of the solution space before presolve and less than 1‰ after presolve in both instances. The size of the non-zero solution space is 36% bigger in the July instance as compared to September and contains 6% more binary variables. The presolve process is completed within 60 sec for both instances and the model components displayed in Table 6.1 are on average reduced by 98.3%.

Figure 6.1 shows the evolution of the upper and lower solution bound of the Fairness model for a 12 hour period for both the July and September instance. A time limit of 12 hours has been chosen, because it enables the scheduler to run the model at the end of a work day and view the output schedule the next morning.
The July instance reaches a solution gap of $\sim 27\%$ after four hours and $\sim 22\%$ after six

[1] Julia documentation: https://docs.julialang.org/en/v1/

[2] Gurobi Optimizer Reference Manual: https://www.gurobi.com/wp-content/plugins/hd_documentations/documentation/9.0/refman.pdf

		Before presolve	After presolve	Percent reduced	Presolve time
July	Rows	870,948	44,966	94.8%	57.58s
	Columns	2,082,156	22,782	98.9%	
	Nonzeros	135,884,039	998,503	99.3%	
	Continous var.	154,656	2,245	98.5%	
	Interger var.	1,927,500	20,537	98.9%	
	Binary var.*	1,921,250	16,516	99.1%	
September	Rows	856,142	45,377	94.7%	53.26s
	Columns	2,088,511	21,890	99.0%	
	Nonzeros	134,422,421	735,490	99.5%	
	Continous var.	152,461	2,062	98.6%	
	Interger var.	1,936,050	19,828	99.0%	
	Binary var.*	1,929,800	15,528	99.2%	

*A subset of the integer variables.

Table 6.1: The value of various model attributes before and after Gurobi presolve for the July and September instances.

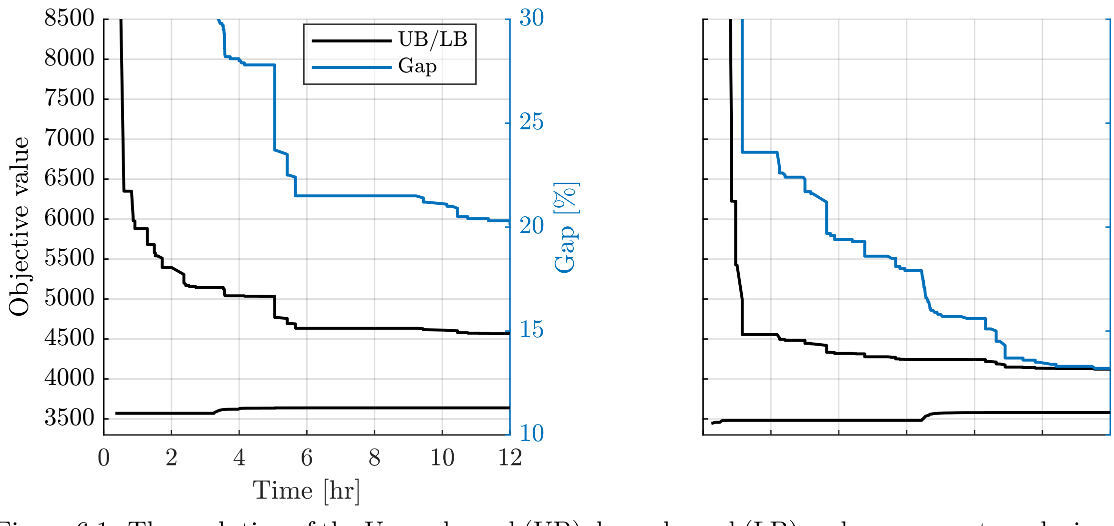

Figure 6.1: The evolution of the Upper bound (UB), lower bound (LB) and gap percentage during a 12 hour period when running the Fairness model on the July (left) and September (right) instances.

hours. Beyond the six hour mark the solution gap of the July instance appears to stagnate. After 12 hours the instance of July reaches a gap percentage of just over 20%.
Within the first hour, the September instance finds a better solution, as compared to the July instance, and obtain a solution gap of around 23%. The solution gap of the September instance continues to converges consistently until the nine hour mark at which the gap is $\sim$ 13%. Beyond the nine hour mark the solution gap of the September instance appears to stagnate.

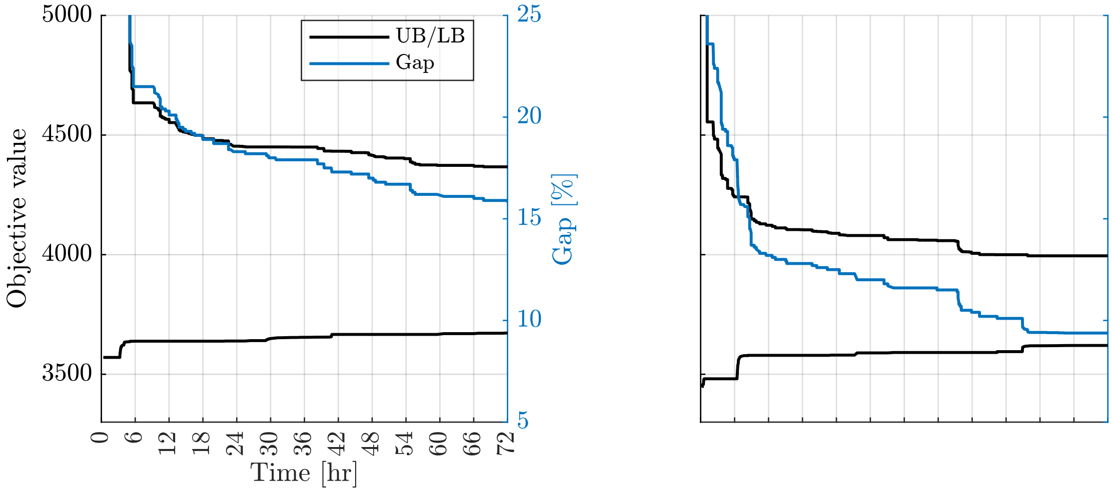

Figure 6.2: The evolution of the Upper bound (UB), lower bound (LB) and optimality gap during a three day period when running the Fairness model on the July (left) and September (right) instances.

From a theoretical perspective, the final solution gap for both schedules is high especially in the July instance. This raises the question of whether the quality of the output schedules are sufficient. In *Practical quality of the Fairness model* we evaluate the schedule optimality from a practical perspective with respect to Key Performance Indicators (KPIs).

To investigate the potential stagnation of the solutions seen in Figure 6.1, the Fairness model was run for three days. In terms of the high solution gaps, we also found it interesting to investigate which of the solution bounds that would improve beyond the 12 hour mark. The result of this experiment is presented in Figure 6.2.

The July instance reaches a solution gap of $\sim 16\%$ after 72 hours. This is 4% less than what was achieved after 12 hours. When considering the entire 72 hour period, the solution gap does not appear to stagnate, and both the upper and lower bounds are improving beyond the 12 hour mark.
The September instance reaches a solution gap of just under 10% which is 3% less than at the 12 hour mark. The solution gap of the September instance also does not stagnate when considering the entire 72 hour period, however after running for 48 hours and 60 hours, respectively, a steep local improvement in solution gap is seen.

To investigate why the two instances converge differently and why the solution gap of the September instance reaches a lower value than in the July instance, the impact of starred wishes are analysed. The number of star-wishes and the type of star-wishes, a work-wish or an off-wish, are one of the main differences between the two instances.
The Fairness model is again run for 12 hours where all stars are removed from all wishes. Thus, no wish needs to be granted as a hard requirement. A starred wish reduces the solution space significantly as the sub solution space for that specific physician and on the day of the wish is limited to the boundaries set by the wish.
First we investigate how much larger the two instances become without starred wishes. This is presented in Table 6.2. When comparing the two instances with no starred wishes, the overall trend is similar to the trends seen in Table 6.1. However, when comparing the

		Before presolve	After presolve	Percent reduced	Presolve time
July	Rows	871,701	93,120	89.3%	80.19s
	Columns	2,082,156	41,880	98.0%	
	Nonzeros	135,896,061	2,634,164	98.1%	
	Continous var.	154,656	2,402	98.4%	
	Interger var.	1,927,500	39,478	98.0%	
	Binary var.*	1,921,250	32,381	98.3%	
September	Rows	856,825	88,046	89.7%	66.42s
	Columns	2,088,511	37,778	98.2%	
	Nonzeros	134,436,179	2,129,299	98.4%	
	Continous var.	152,461	2,331	98.5%	
	Interger var.	1,936,050	35,447	98.2%	
	Binary var.*	1,929,800	28,601	98.5%	

*A subset of the integer variables.

Table 6.2: The size of the model before pre-solving and after pre-solving with respect to the rows, columns, non-zeros, continuous variables, integer variables and binary variables for July and September with no wishes.

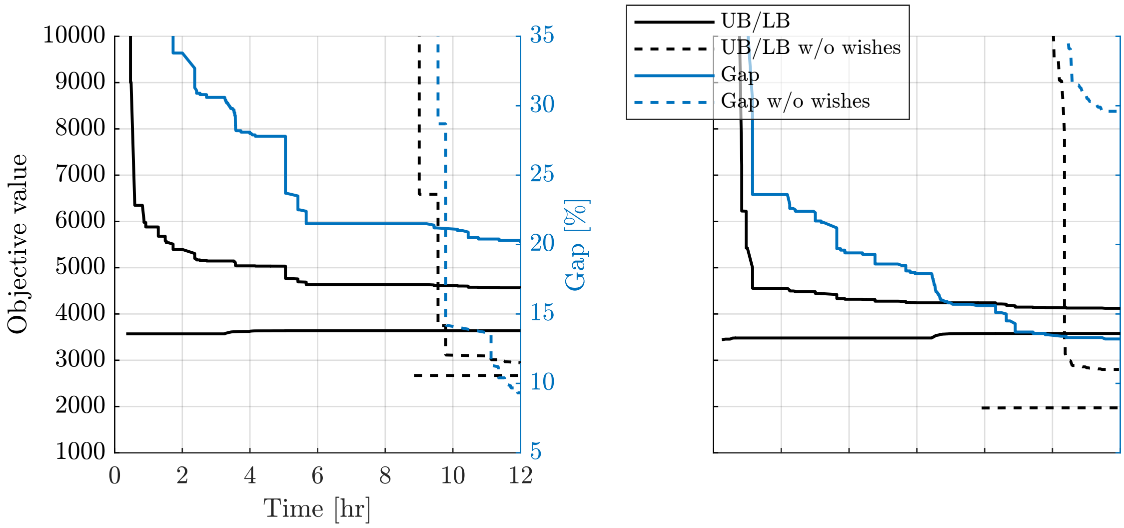

Figure 6.3: The evolution of the Upper bound (UB), lower bound (LB) and optimality gap during a 12 hour period when running the Fairness model on the July (left) and September (right) instances. Each plot pictures both the full instances and the instances where stars has been removed from all wishes.

same instance, with and without starred wishes, the resulting model attributes increase significantly. The number of non-zeros increase by 164% and the number of binary variables increase by 96% in the July instance when stars are removed from all wishes. In the September instance, the number of non-zeros increase by 190% and the number of binary variables increase by 84% when removing stars on wishes.

The solution bounds for both instances, with and without starred wishes, are pictured together in Figure 6.3. It takes 8-9 hours before the Fairness model finds an initial feasible solution in both the July and September instance when stars are removed from all wishes.

However, once an initial solution is found, the upper bound improves much faster than in the instances with starred wishes. In the July instance without starred wishes a solution gap of less than 10% is reached within the 12 hour time limit, even though the first feasible solution is found after 9 hours. When including stars on wishes, the first feasible solution is found within the first hour but the solution gap is still 20% at the time limit. The solution gap for September is just under 30% at the 12 hour mark when removing all stars, however, the first feasible solution is also found approximately an hour later in September as compared to the July instance without starred wishes.

6.2 Practical quality of the Fairness model

The best solution found by the *Fairness model* is post-processed in Julia and saved as a series of raw Excel sheets. Using conditional formatting rules, the raw data is automatically transformed into an output of 11 sheets which provide the scheduler both the final schedule as well as supporting statistics. Ideally, the output would be an interactive interface that would update according to adjustments made my the scheduler. However, for the sake of a proof-of-concept, static Excel sheets sufficed for this thesis. The content of each of the sheets available to the scheduler is described below and visualised in Appendix B.0.1 for a single instance.

Schedule is the presentable schedule of assigned shifts and add-ons including distinct coloring of wishes granted and not granted. An example of this sheet can be seen in Figure B.1.

Wish Status details all staff wishes including if the wish is star marked. Not granted wishes are highlighted with red. An example of this sheet can be seen in Figure B.2.

KPIs includes a summary of the most important indicators such as over/under hours, the overview of wishes granted and more. Most KPIs detail a minimum, average, and maximum value. An example of this sheet can be seen in Figure B.3.

Demand details the planned demand vs. the actual assigned demand. An example of this sheet can be seen in Figure B.4.

Hours and contract consist of everything time related for each physician and their contract. Additionally, it provides the number of weekends a physician is working. An example of this sheet can be seen in Figure B.5.

Rest periods details the number of both short and long rest periods assigned on each day of the schedule. An example of this sheet can be seen in Figure B.6.

Time to next working shift is the time, in hours, to the next working shift on each day. Days on which a physician is off are marked as 'no shift'. An example of this sheet can be seen in Figure B.7.

Number of shifts/add-ons pr. day details the assigned number of each shift and add-on on each day. An example of this sheet can be seen in Figure B.8.

Number of shifts/add-ons pr. physician details the total number of each shift and add-on assigned to each physician in the entire schedule. An example of this sheet can be seen in Figure B.9.

Historic shift data shows the updated number of specific shifts assigned to a physician within a norm period, e.g. the number of compensation days used in the current year. These specific counted shifts are defined in *Data sheet 12: History and Targets*. An example of this sheet can be seen in Figure B.10.

Fairness on counters details the number of shifts/add-ons assigned for each fairness counter to each physician in relation to their number of available working days in the month. An example of this sheet can be seen in Figure B.11.

The 11 sheets provides a detailed picture of almost all elements of the final schedule. The scheduler can utilize some of the sheets for a quick overview and other sheets for a detailed deep dive into specific elements.

This analysis includes three studies of the practical quality of the schedules generated by the *Fairness model* in 12 hours. The first study examines KPIs with respect to minimum, average, and maximum values for July and September. The second study considers the 'costs' of adding fairness elements to the model by comparing the difference in selected KPIs between the output schedule of the *Fairness model* and the *MVP model* (the MVP model was run for 12 hours as well). The last analysis showcases how well the fairness elements work by comparing KPIs, which should be affected by fairness, in the output schedule of both the Fairness and the MVP model.

The first study is conducted to present an overview of the practical quality of the two schedules (July and September) and to compare them. The KPIs inspected are presented in Figure 6.4.
In the schedule of July, the average amount of over hours is around nine whereas in September it is just above two hours. The maximum amount of over hour in July is ∼ 22 hours and ∼ 7 hours in September. The difference in over hours may be due to July being a holiday month with many physicians partly absent. The schedule of July has no under hours which could indicate there is a shortage of physicians as compared to the

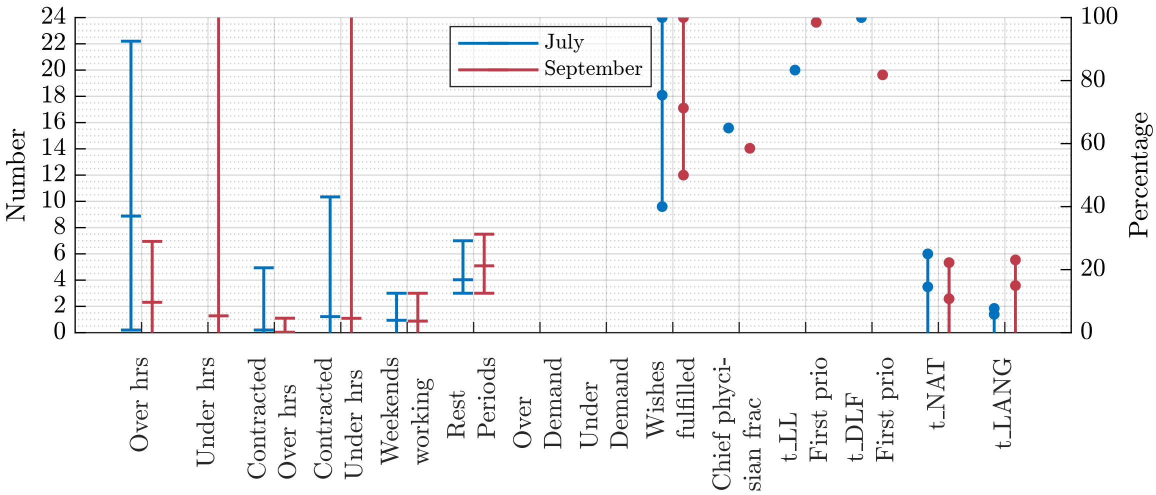

Figure 6.4: Minimum, average and maximum values for the set of KPIs obtained when running the Fairness model for 12 hours on both the July and September instance. The KPIs displayed with line whiskers are connected to the left y-axis [number] and the KPIs displayed with circular whiskers are connected to the right y-axis [percentage].

required demand. The average amount of under hours in September is one hour. This is due to the fact that two out of the fifty physicians have 21.05 and 25.8 under hours, respectively, as the only two physicians with under hours. The reason for this, is that these two physicians have wished for several week 'off-days' (Mon-Fri) with a star. Moreover, the two physicians do not have the competencies it requires to man either weekend shifts or night shifts. Thus, they cannot make up for the hours they should have worked on the days they wished for being 'off'. It is likely that these starred 'off-day' wishes are actually compensation days or similar.
The average values for over and under hours on contracts are approximately the same, close to zero and one hour, respectively, in July and September. The maximum amount of contracted under hours for September is 32 hours for physician CSV. CSV is on a special full time contract which only includes a limited set of shifts. CSV is assigned regular day shifts 'D' in the clinic (not in his contract) 20% of his working time. This can be due to a shortage of available physicians or the fairness element associated with the clinical day shift 'D8' (day shift plus added responsibility) which CSV is allowed to man.
The average number of working weekends is around one and the maximum is three. The reason that some physicians are assigned work on three weekends can either be because they have wishes for it or because the punish value associated with working two subsequent weekends is too low in comparison to other objectives. The average number of rest periods assigned is above the target in both instances and there is no shortage or surplus for any minimum or exact demands, respectively. The average percentage of granted wishes is a little lower in September, around 70%, as compared to July, but the worst case percentage is higher in September than in July. The fraction of chief physicians who man night shifts/add-ons is around 60% for both schedules, which is below the maximum allowed fraction of 2/3.
Some counters can be satisfied by multiple shifts, but some of these shifts might be preferred over others. We denote these *prioritized counters* and the shifts that satisfy a prioritized counter are ordered by priority. In both July and September over 80% of the demand on the two priority counters, t_LL and t_DLF, is satisfied by the first priority shift. In terms of fairness on undesired shift types, such as long shifts an night shifts, the schedules of July and September are also similar. The average percentage of night shifts assigned, as compared to the number of days a physician is available to work, is $\sim 15\%$ and the worst case is $\sim 20\%$. In July the maximum percentage of long shifts assigned is less than the average percentage in September. This is because July is a holiday month where the demand for long shifts is reduced.

The second study in this analysis evaluates the 'cost' of adding fairness to the model. This is done by comparing the average value of certain KPIs with and without adding the fairness elements introduced in the *Fairness model*. Hence, the KPIs are evaluated by running both the Fairness model (full model with fairness) and the MVP model (full model without fairness) for 12 hours on the instance of July only for simplicity.
The objective value of the MVP model is 2663 after 12 hours, whereas the part of the objective in the Fairness model which corresponds exactly to the objective of the MVP model, is 3619 after 12 hours. Thus, all but the fairness objectives increase by a total of 35% when adding fairness to the model. In Figure 6.5 the average values of a set of KPIs, that could potentially be affected negatively by added fairness elements, are presented. All the presented KPIs worsen when fairness elements are added except over hours with respect to contracts. General over hours and the granted fraction of wishes are the two KPIs which appear to worsen the most as a result of adding fairness. When adding fairness, physicians are on average assigned two more over hours which corresponds to 100 additional over hours in total. The average percentage of granted wishes decreases by

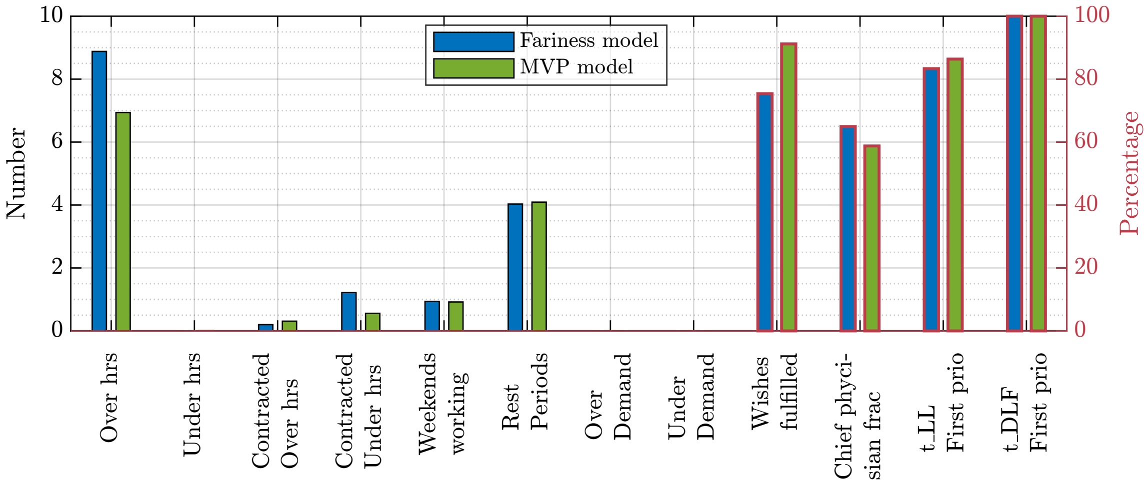

Figure 6.5: Average value for a set of KPIs obtained when running both the Fairness and MVP model for 12 hours on the instance of July. The KPIs displayed with bars that are outlined with black are connected to the left y-axis [number] and the KPIs displayed with bars that are outlined with red are connected to the right y-axis [percentage].

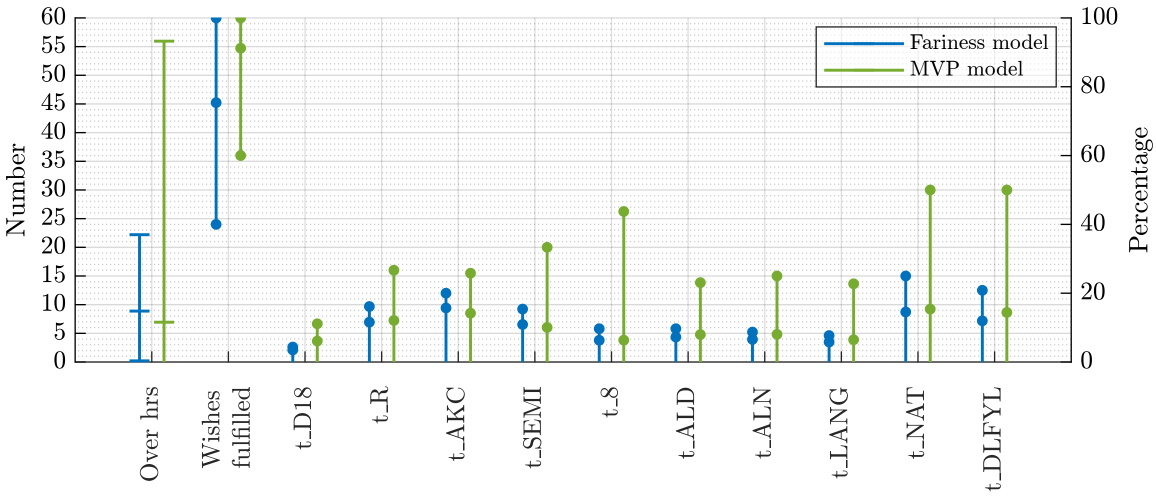

Figure 6.6: Minimum, average, and maximum values of fairness KPIs when running both the Fairness and MVP model for 12 hours on the instance of July. The KPIs displayed with line whiskers are connected to the left y-axis [number] and the KPIs displayed with circular whiskers are connected to the right y-axis [percentage].

∼ 10% when adding fairness. This corresponds to fulfilling 23 less wishes. Even though the average performance of the KPIs are better without than with fairness elements, a crucial part of fairness is to minimize the variance within the KPIs rather than optimizing the averages. Therefore, the third study of this analysis investigates if the added fairness elements improve the variance of the elements of a schedule that should be fair.

Fairness is applied to over hours, percentage of granted wishes, and certain counters such as 't_NAT', i.e. night shifts. In Figure 6.6, the KPIs related to these elements are shown for both the full Fairness model and for the MVP model which has no fairness.
The only fairness KPI that does not appear to become more fair when adding fairness is the percentage of fulfilled wishes. The difference between the average percentage and the worst case percentage, i.e. the variance, is around the same both with and without added fairness. However, because the position of the lower quartile is not visualized, one cannot

rule out that the variance of the percentage of fulfilled wishes could still have improved for the majority of the physicians. All other fairness KPIs can be seen to become more fair in the model with fairness.
Considering the amount of over hours, when adding fairness, the worst case scenario is reduced from approximately 56 hours to 23 hours. This corresponds to approximately four less shifts in one month. A similar improvement in fairness is also observed in almost all counters shown. For example, the maximum percentage of night shifts assigned (counter 't_NAT') is reduced from 50% to around 25%, and the maximum percentage of long shifts (counter 't_LANG') assigned, is reduced from over 20% to less than 10%.

6.3 Parameter study of the Two Stage model

A lexicographic optimization approach was introduced in the *Two stage model* as an attempt to try to speed up the solution time or improve solution bounds within a set time frame. However, there are two elements of the Two-stage model which have a significant impact on the performance of the approach, namely: the target optimality gap in the first stage, and by how much the first stage objective value is allowed to 'worsen' in the second stage, hereinafter denoted as the 'slack'. The target gap in the first stage is a cost benefit consideration. A good solution should be found, but it should not take up the majority of the solution time. Moreover, as it was seen in *Practical quality of the Fairness model*, adding fairness affects the optimality of the objectives optimized in the first stage. Thus, if the slack value must be high in order to obtain a good compromise between fairness objectives and the remaining objectives, then it does not make sense to spend time achieving a small solution gap in the first stage.

The parameter tuning of the Two stage model is done using both the instance of July and September. Ideally a much larger set of instances should be used to obtain good global parameter estimates. The tuning experiments are set up to run for three hours only as the purpose is to try to obtain better solutions quicker. The optimality gaps tested in the first stage are; 5%, 10%, and 15% and the slack percentages tested in the second stage are; 10%, 15%, and 20%. Thus, a total of nine combinations of parameters are tested for each month.
The experiments for July are presented in Figure 6.7 and the experiment for September are presented in Figure 6.8. To visually compare the two months, a third plot has been constructed with a zoomed in view of both plots side-by-side, zoomed in at the end of the three hour run time. This plot is presented in Figure 6.9. The values shown in Figure 6.7, 6.8, and 6.9 are artificially constructed. In the second stage of the algorithm, only the fairness elements are optimized. Thus, while the solver searches for an optimal solution only the value of this objective is available and not the value of the objectives from the first stage. In the worst case, the combined value of the objectives from the first stage utilize all the available slack. We choose to add this constant conservative value to the objective values from the second stage. Thus, why the values in the mentioned plots are artificially constructed. In all experiments, it was found that the full slack was indeed utilized in the best found solution within the time limit.

For the instance of July pictured in Figure 6.7, it can be seen that stage one takes under 15 minutes to complete for a 10% and 15% optimality gap and just under an hour for a 5% optimality gap. During all of the second stage, the three 5% optimality gap experiments perform worse than most of the other experiments. At the end, four experiments ends up at almost the same best found objective value. These are; a 10% gap combined with both 15% and 20% slack, as well as a 15% gap combined with both 15% and 20% slack.

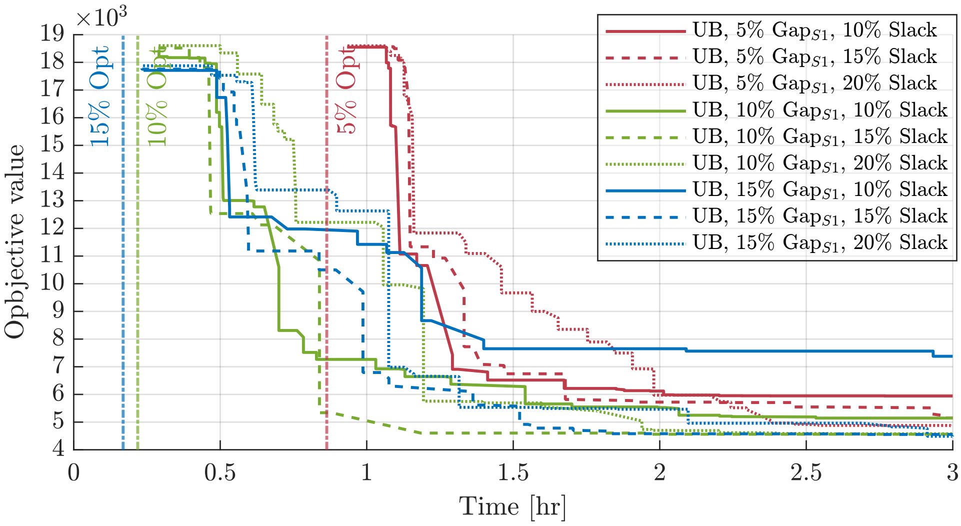

Figure 6.7: Upper bound comparison of the nine settings tested as part of the parameter tuning of the optimally gap in stage one and the slack in stage two of the Two Stage model. Here pictured for the instance of July.

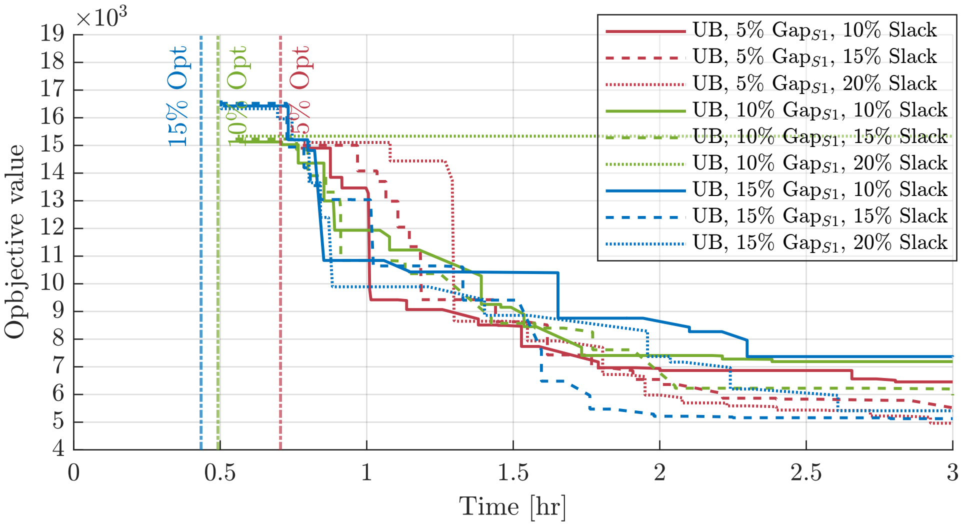

Figure 6.8: Upper bound comparison of the nine settings tested as part of the parameter tuning of the optimally gap in stage one and the slack in stage two of the Two Stage model. Here pictured for the instance of September.

For the instance of September pictured in Figure 6.8, it can be seen that stage one takes under 30 minutes to complete for a 10% and 15% optimality gap and around 45 minutes for a 5% optimality gap. At the end, four experiments ends up close to the same best found objective value. These are; a 5% gap combined with both 15% and 20% slack, as well as a 15% gap combined with both 15% and 20% slack. An odd observation is the experiment with a 10% optimality gap and 20% slack. This experiment stagnates immediately at the beginning of the second stage. The two other experiments which are also set to a 10% optimality gap do not stagnate in the second stage. What is even more odd is that the 10% optimality gap and 15% slack experiment initially finds almost twice as many nodes as the 10% optimality gap and 20% slack experiment does, in the second stage. Yet, the 10% optimality gap and 15% slack experiment has no convergence issues in the second stage.

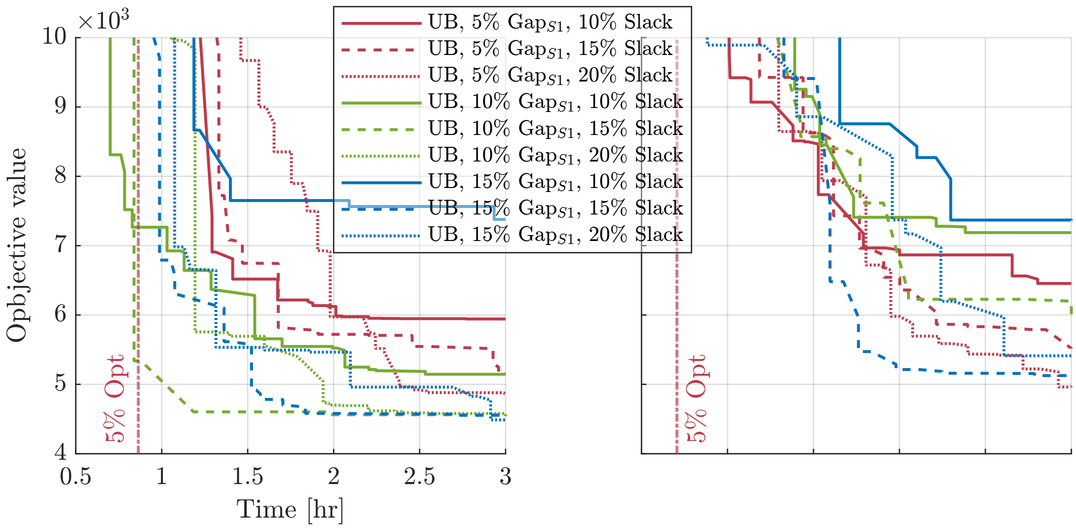

Figure 6.9: Zoomed in view of Figure 6.7, left, and Figure 6.8, right. Generally speaking, the trend seems to be that the higher the combined slack (gap and slack combined), the better the objective becomes.

From Figure 6.9 a general trend can be seen for both July and September. The higher the slack percentage, the better the total objective value is at the three hour mark. For example, comparing all the 5% optimality gap experiments with each other, the 20% slack performs best followed by the 15% slack and lastly the 10% slack.

The experiment with a 15% optimality gap and 15% slack performs best overall when considering both instances. In July this setting find one of the best objective values and it finds this solution as the second fastest of all nine experiments. Moreover, in September it converges to a good solution fastest out of all of the nine experiments and is only surpassed at the very end of the three hour mark by one other experiment. Thus, a 15% optimality gap and a 15% slack seems like the best global setting, and this setting is used in the further analyses.

6.4 Two Stage model versus the Fairness model

This analysis is conducted to compare the performance of the *Two stage model* with the *Fairness model*. The Two Stage model is run with a 15% optimality gap in stage one and 15% slack in stage two, since this was concluded to be the best setting in *Parameter study of the Two Stage model*. The two solution approaches are compared for both July and September and are run for 12 hours. The result can be seen in Figure 6.10. As the combined objective value of the Two Stage model (the sum of stage one objectives and stage two objectives) can only be determined once the solver exits the optimization process, the Two Stage model was run with multiple time limits. Each run is marked with a dot in Figure 6.10 and the red straight lines shown between the dots are interpolations. For the same reason, it is only the upper bounds which are compared in this analysis.

The Two Stage model surpasses the Fairness model after around one hour and 30 minutes in July. At the 2 hour mark it has a significant better objective value of around 15% more. At the 3 hour mark the solution found by the Two Stage model is better than the solution found by the Fairness model after 12 hours. In September however, the Fairness model completely outperforms the Two Stage model. Within the first two hours the Fairness model has found a solution which is better than any solution found by the Two Stage in all of the 12 hours considered.

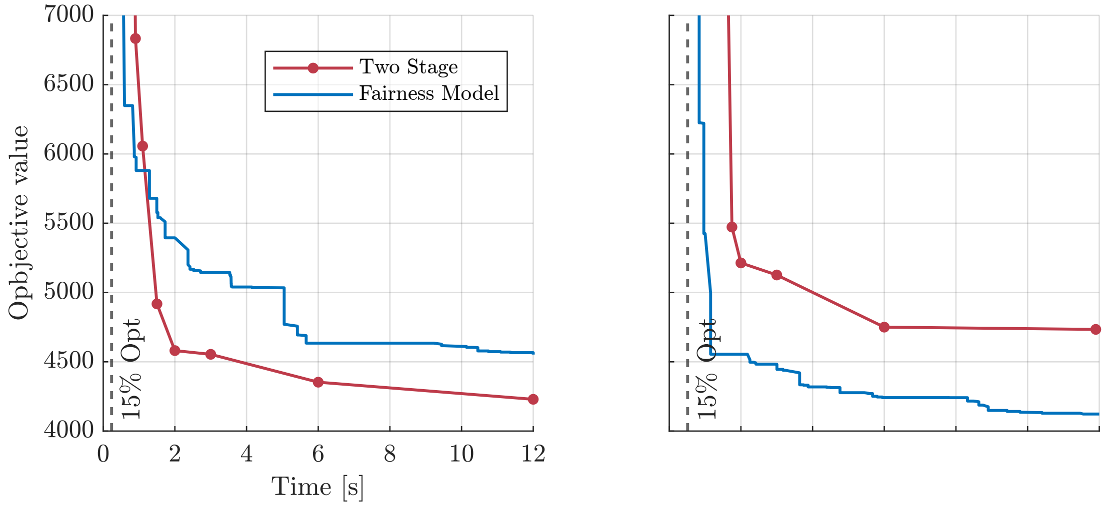

Figure 6.10: The upper bound of the Two stage model and the Fairness model for July (left) and September (right). In the July instance the Two Stage model outperforms the Fairness model and vice versa in the September instance.

6.5 Fairness model with warm start

Despite the inconsistent performance of the Two Stage model in *Two Stage model versus the Fairness model*, its general convergence trends in the second stage seemed faster than for the regular Fairness model. Consequently, it was chosen to test a warm start approach on the full *Fairness model*.

The warm start solution provided to the Fairness model is the solution of the *MVP model*, i.e. equivalent to the stage one solution in the *Two stage model*. The warm start solution is optimized to 15% optimality before being passed as an initial solution to the Fairness model. This experiment was conducted for both July and September and was run for 6 hours, as the main idea was to test if the warm start approach gave an early advantage in terms of bounds. The lower bound, upper bound, and optimality gap of the Fairness model and the Fairness model with a warm start are shown in figure 6.11 side-by-side.

In July, the the Fairness model with a warm start has a lower optimality gap than the regular Fairness model throughout the entire run-time. This is because the Fairness model with a warm start manages to lower its upper bound much quicker than the regular Fairness model. In September, the regular Fairness model has a $\sim 3\%$ better optimality gap than the Fairness model with a warm start throughout the entire run-time. However, when comparing the upper bound, the two solution approaches are almost identical after 2 hours and onwards.

For comparability, the upper bound of the three tested solution approaches; the regular Fairness model, the Two stage model, and the Fairness model with warm start, are presented side-by-side in Figure 6.12. In the July instance, the Two Stage Model and the Fairness model with a warms start both find solutions of similar quality in a similar amount of time. The regular Fairness model performs worse in terms of solution quality and speed. In the September instance it is the regular Fairness model and the Fairness model with a warm start which both find solutions of similar quality in similar amounts of time. Here, the Two Stage model performs worse in terms of solution quality and speed.

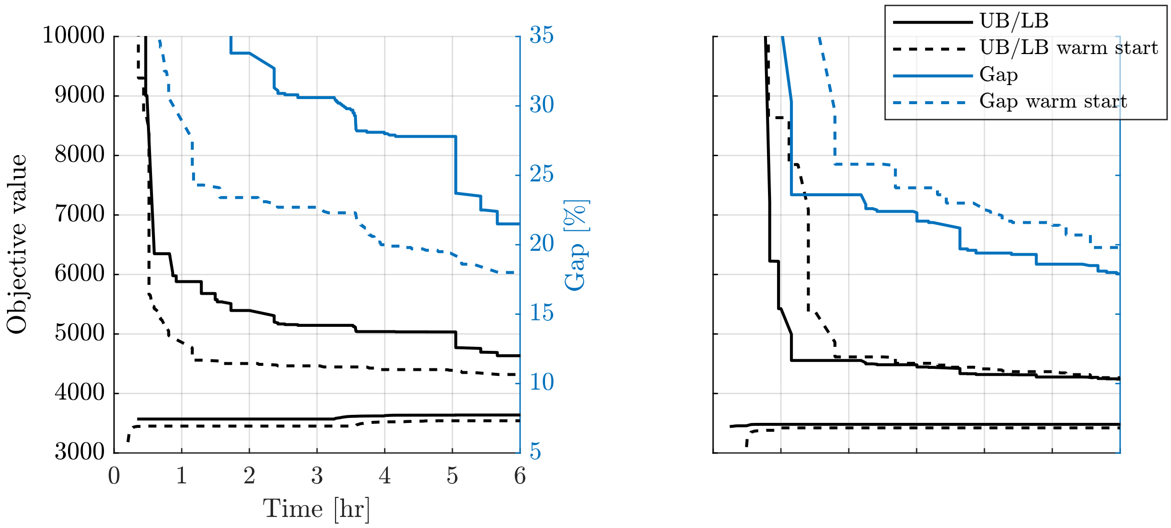

Figure 6.11: The upper bound, lower bound, and the optimality gap of the Fairness model with and without warm start for July (left) and September (right). The Fairness model with warm start obtains a better upper bound in the July instance and obtains almost the same bound in the September instance as the regular Fairness model.

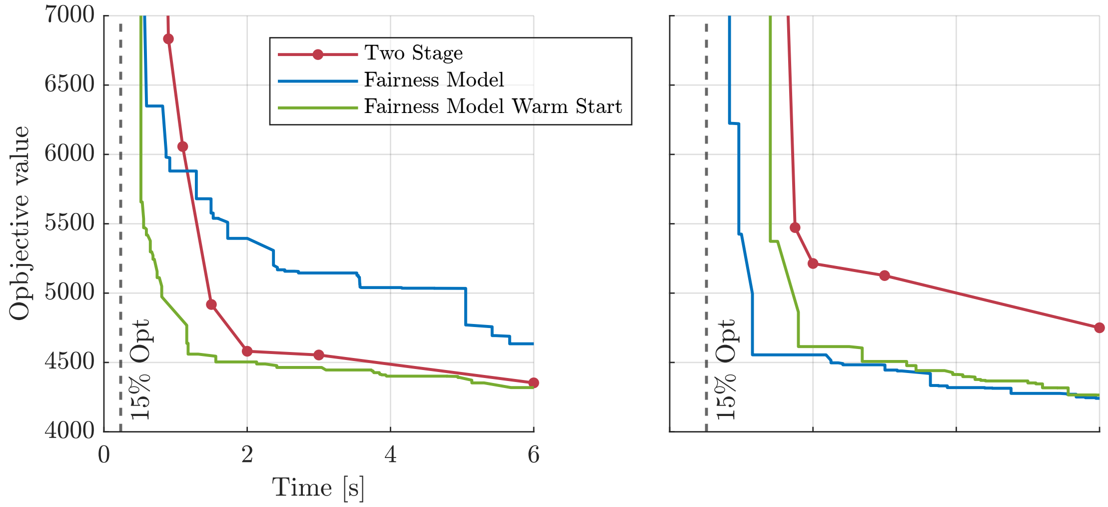

Figure 6.12: The upper bound of the Two Stage model, the Fairness model without warm start and with warm start for July (left) and September (right). The performance of the Two Stage model and the Fairness model is ambiguous with respect to both instances whereas the performance of the Fairness model with warm start is stable.

6.6 Decomposition of the MVP model

Since the *Decomposition model* does not include fairness elements it is initially compared to the *MVP model* which it is equivalent to. The comparison considers the evolution of the upper bound value with respect to the solution time of each model and is pictured in Figure 6.13 for the month of July. The Decomposition model has been optimized to optimality in the master and all subproblems. Within 10 minutes, the MVP model stagnates at an objective value of around 3000 whereas the Decomposition model reaches a solution just above 4000 within the 6 hour time limit considered. The same trend is apparent for the month of September although this comparison has been left out of this analysis.

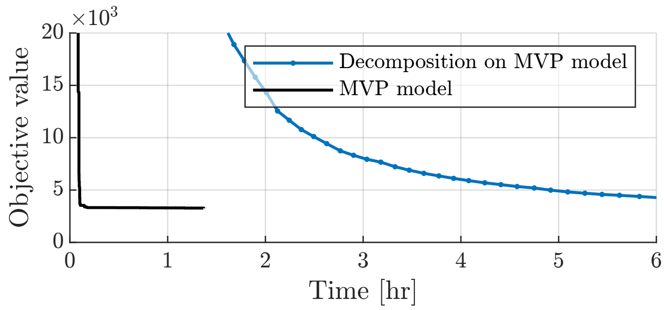

Figure 6.13: The evolution of the objective value of the MVP model with and without Dantzig-Wolfe decomposition. The regular MVP model outperforms the decomposition approach in terms of both time and solution quality.

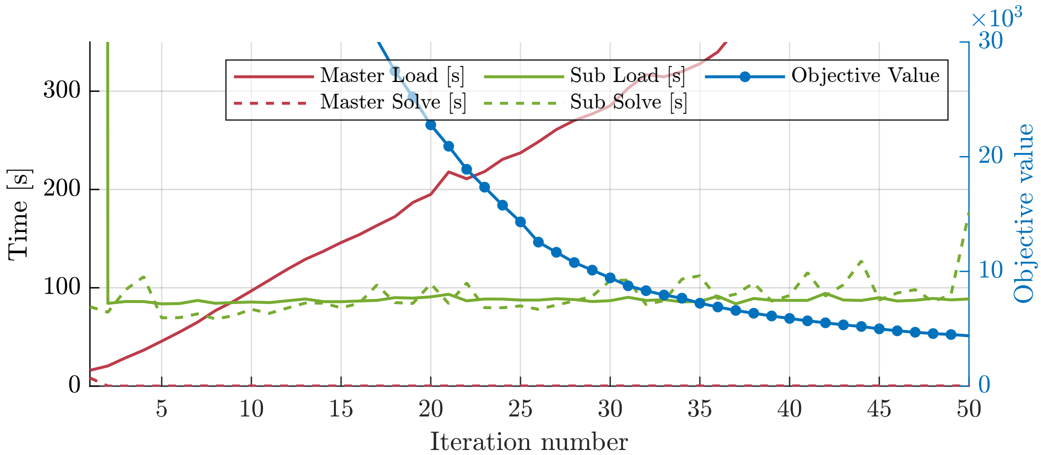

Figure 6.14: A break-down of the time spend on each element of the MVP model with Dantzig-Wolfe decomposition solution approach (left axis) as well as the evolution of the objective value in the process (right axis). The time it takes to load the master problem increases linearly with each iteration whereas the other model elements remain approximately constant with each iteration.

An in-depth time study of the Decomposition model was conducted on July to identify which elements accounted for significant parts of the solution time. The model was set to complete 50 iterations in which each of the four solution elements were timed; master problem load time, master problem solve time, subproblems combined load time, and subproblems combined solve time. Additionally, the objective value was also measured to see the progress in each iteration. The time study is pictured in Figure 6.14.
The combined loading time of all subproblems is consistent throughout all iterations and takes around 90 seconds. This alone results in a total time of one hour and 15 minutes for the first 50 iterations. The load time of the master problem increases linearly with the addition of new patterns in each iteration. The load time of the master problem takes more time than the combined load time of all subproblems in the eighth iteration already. However, the solution time of the master problem is insignificant in all iterations. Lastly, the combined solution time of the subproblems is also consistently ~ 90 seconds in each iteration. Thus, the combined solution time of the subproblems alone by far exceeds the solution time of the MVP model.

6.7 Analysis of the optimality gap from a practically perspective

This analysis compares KPIs with respect to minimum, average, and maximum values when running the *Fairness model* with a warm start, cf. *Fairness model with warm start*, for two and 12 hours. This study is conducted for both the instance of July and September, pictured in Figure 6.15 and Figure 6.16, respectively. The final upper bound, lower bound, and optimality gap obtained for the four runs are shown in Table 6.3. For July, the upper bound improved by 196 when running the model for 12 hours instead of two hours, which is equivalent to a 5.6% difference in optimality gap. For September, the upper bound improved by 663 when running the model for 12 hours instead of two hours, which is

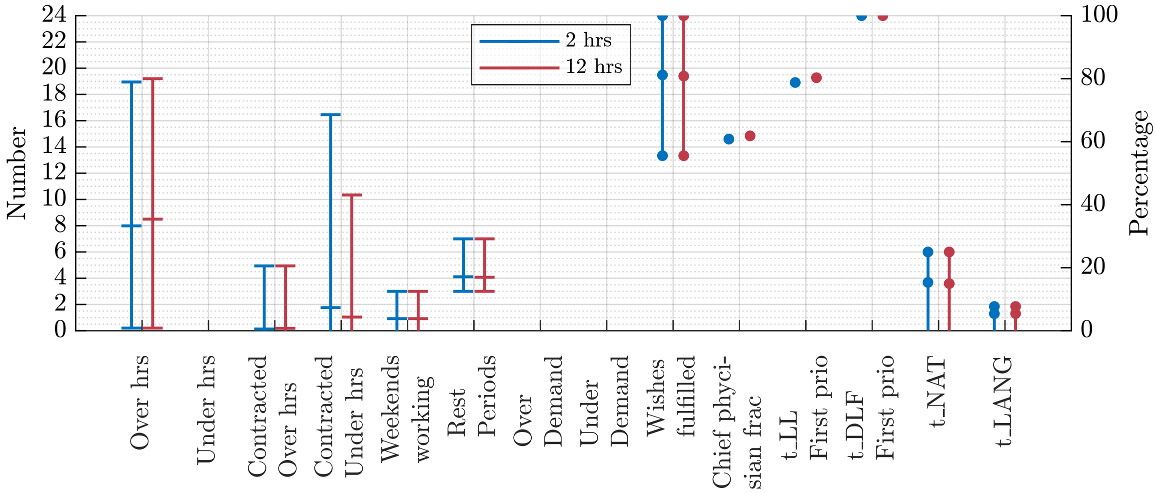

Figure 6.15: KPIs for the July instance when running the Fairness model with warm start for two and 12 hours. The KPIs displayed with line whiskers are connected to the left y-axis and the KPIs displayed with circular whiskers are connected to the right y-axis. Minor improvements are achieved in the 12 hour run as compared to the 2 hour run.

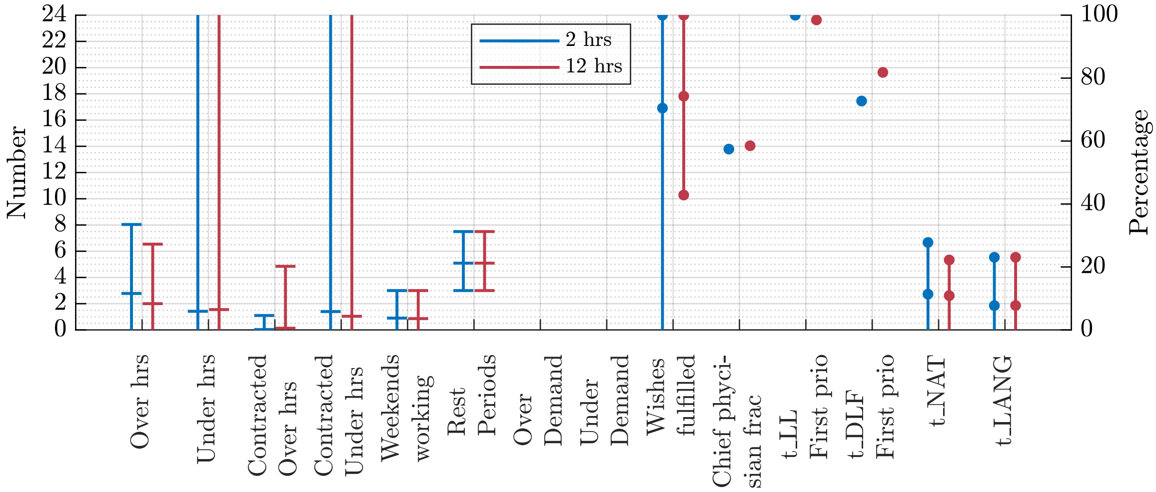

Figure 6.16: KPIs for the September instance when running the Fairness model with warm start for two and 12 hrs. The KPIs displayed with line whiskers are connected to the left y-axis and the KPIs displayed with circular whiskers are connected to the right y-axis. Generally, minor improvements are achieved in the 12 hour run as compared to the 2 hour run, expect for a significant improvement in the worst case scenario for *wishes fulfilled*.

	Hours Run	Upper Bound	Lower Bound	Optimality Gap
Jul	2	4504	3452	23.4%
	12	4308	3541	17.8%
Sep	2	4745	3405	28.2%
	12	4082	3465	15.1%

Table 6.3: The upper bound, lower bound, and optimality gap for the instances of July and September. Here, all values are shown for both a run-time limit of two and 12 hours.

equivalent to a 13% difference in optimality gap. Thus, in September the difference in the quality of the KPIs in the two hour run and in the 12 hour run should be more noticeable than in July. However, since not all of the objectives are considered in this analysis this trend is not guaranteed to be visible here.

First and foremost, all demand is met and no critical elements are unfulfilled in all four runs considered. In the instance of July, there are some KPIs, such as the amount of over hours, which are better after the two hour run as compared to the 12 hour run. However, for the majority of the KPIs pictured in Figure 6.15, there are either no noticeable differences between the two runs or the KPI of the 12 hour run is slightly better than in the two hour run. Overall, the difference between the two runs for the instance of July seems close to insignificant from a practical standpoint.

The KPIs for September are shown in Figure 6.16. The 12 hour run performs slightly better in the majority of the KPIs than the two hour run. For the instance of September the most significant difference between the two time limits is seen in the fairness of the percentage of fulfilled wishes. In the two hour run there is a physician who gets 0% of her/his wishes fulfilled whereas the minimum percentage of fulfilled wishes is just above 40% in the 12 hour run.

7 Discussion

This chapter entails essential considerations of the work carried out in this thesis. The discussion concerns the data structure, key takeaways from the analyses, applications of the model, the general model approach and further work.

7.1 Data structure

The idea behind the chosen data structure was to make it as simple as possible for the scheduler to maintain. Still, the data structure needed to be complex enough to accommodate changes in contracts, demand, competencies, local regulations, etc. One of the main data considerations was how to structure shifts and add-ons. An add-on is always assigned in combination with a shift, thus, instead of treating shifts and add-ons as two separate components, all combinations could be generated and treated as a unique set of shifts. By only having a single list of shifts, most schedulers would in our opinion, find it easier to update and maintain counters and wishes. This is because the user does not have to consider the various shift-add-on combinations, but instead only needs to select shifts from a single list. However, by merging shifts and add-ons, multiple shifts will exist in various versions, where the only difference should be the information related to the add-on. Thus, in terms of maintenance, changes to a single shift must be duplicated to all its variants. In general, adding new shifts, competencies, counters, etc. will require more work for the scheduler, as the size of the data set will be larger when all shifts and add-ons are merged. We deduced that the likelihood of data errors would be reduced by limiting the size of the data.
When complexity is removed from the data, typically the complexity of the model will increase in order to maintain the same functionalities. Modelling add-ons as an individual component results in an additional dimension in the model. In hindsight the best solution may have been to add a data processing step prior to the model formulation. The data processing step should as a minimum merge all shift-add-on combinations.

One limitation we were able identify with the chosen data structure, is the ability to define counters. Because a counter is defined as all possible combinations of a set of valid shifts and a set of valid add-ons, it is only possible to select specific subsets of all shift-add-on combinations. As an example, if a scheduler wants to define a counter for all nightly shift-add-on combinations the following problem arises: the combination of any day shift and a night add-on should be counted, but the combination of a night shift and with the 'no add-on' should also be counted. This also allows the combination of any day shifts and the 'no add-on' to be counted, despite not being a nightly shift-add-on combination. We found this limitation to be an issue in our work, because night shifts and add-ons should be evenly distributed in terms of fairness. We tested a solution in which we constructed a new type of counter which is the sum of a set of regular counters. We tried to apply fairness to this new counter which worked well. This addition has not been integrated in the model nor the data structure presented in this thesis.

7.1.1 Scaleability and controls

In general the data structure is set up such that new employees, shifts, competencies, counters, etc. can easily be added or removed. The data structure provides additional less intuitive elements of flexibility. Some example of these are:

1. If a shift should only be assigned to a single or a subset of physicians, a new competence can be added as a shift requirement and only given to the desired physicians.

2. If a counter requires two or more competencies, e.g. a physicians which is both a pediatrician and a chief physician, then, a new competence combining the two can simply be added and act as the competence requirement for the counter. The same applies in cases with 'either or' competencies, e.g. either a chief physicians or a senior registrar.

One flexibility element that has not been included in the model, is the option to specify weekly demand or contracted days. In the anesthesia department, some physicians are contracted to a weekly administrative day. Currently, the model can only assign a certain number of contracted shifts on a monthly basis, i.e. it is not guaranteed that these shifts will be evenly distributed over each week. A work shift can of course always be starred and is thus guaranteed to be assigned on a certain day.
Overall, we postulate it should be possible to change the data and use the same model for different departments at Rigshospitalet, given the assumption that the same set of regulations apply.

Parameters associated with rules and regulations have been set up on a personal level such that local agreements can easily be added without affecting everyone. A current limitation is that local agreements are not punished in the objective. This is an issue if a physician, out of the ordinary, has agreed to working e.g. 8 days in row instead of 6. Although the physician has agreed to it, this exemption to the regular rule should only be exercised as a last resort.
Parameters associated with targets have also been set up on a personal level. This allows the scheduler to manipulate the output of the model in various ways. Some examples of this are:

1. If a physician is behind or ahead on e.g. total working hours or the number of rest periods within a norm period, the scheduler can adjust the physician's targets to reflect this.

2. The scheduler can adjust a punish value associated with the over hours for each of the physicians. It is three times as expensive for the model to assign over hours to a physicians with an over hour status of three, than to a physicians with an over hour status of one. This allows the scheduler to decrease the likelihood that certain physicians get over hours.

Thus, in general the data structure is set up to provide the scheduler flexibility with respect to various elements. This further improves the possibility of scaling the model across multiple departments.

7.2 Model analysis

In this section the results detailed in *Analysis and results* are discussed. Firstly, the discussion of the theoretical and the practical quality of the model output is presented. Secondly, the discussion of the various solution speed-up approaches is presented. Lastly, a summary and an overall analysis evaluation is presented.

7.2.1 Quality assessment

From a theoretical perspective, the optimality of the model is far from optimal within a 12 hour time limit, above 10% for all instances tested. We extended the run time to 72 hours to determine if the model simply stagnated or if perhaps the solution was near optimality, hence only the lower bound would continue to improve. It was seen in the analysis that during the entire 72 hours both solution bounds continued to improve. Thus, from a theoretical point of view solutions can still potentially be improved. It is difficult to conclude what the primary cause of the optimization time is. It could be the size of the solution space, the number of integer and binary variables in model, or the complexity of the constraints.

From a practical perspective, various KPIs were analysed in *Practical quality of the Fairness model* and assessed by the anesthesia scheduler. The KPIs shown in 6.4 were for the most part satisfactory. The different targets and demands were not violated and the trade-off between both positive and negative soft elements such as personal wishes and undesired shift types seems to be balanced well.
The fairness of the solution was analysed from two perspectives. The 'cost' of fairness was visualized in Figure 6.5 and the benefits of fairness was visualized in Figure 6.6. When adding fairness, the original objective worsens by a total of 35%. Although this seems significant, Figure 6.5 showed that when spread out across 50 physicians and multiple weighted objectives, no single objective worsened unreasonably. However, from a practical perspective the fairness elements have a crucial impacts for the roster of the individual physician. Working 55 over hours in a single month when the department average is only $\sim$ 7 hours, is unacceptable. Similarly, is it also unacceptable to receive a plan which consists of 50% night shifts when the average is $\sim$ 16%. Logically, we conclude that the benefits of fairness outweigh the costs. Moreover, the scheduler at the anesthesia department also expressed that ensuring a balanced satisfactory level across all physicians was a key component of a good plan.

In *Analysis of the optimality gap from a practically perspective* the difference in practical quality between running the model for two and 12 hours was outlined for the Fairness model with a warm start approach. In the two instances considered, the overall quality of the KPIs were almost identical for a time limit of both two and 12 hours. However, significant outliers are more likely to occur with the two hour time limit. As an example, in the September instance a physician was not granted a single wish. Nevertheless, the average percentage of granted wishes was still found to be almost as good as in the 12 hour run. Since the idea is that the scheduler will always be needed to review and potentially make modifications, the two hour solution may be sufficient. Fixing a few outliers may in some cases be worth the flexibility gained from being able to obtain a solution within two hours.

7.2.2 Solution methods

In the process of developing the model, it quickly became apparent that the optimization process was long. Additionally, just the load time of the model was also a significant. After the last fairness constraints were added to the model, the load time was close to two hours. Therefore, the first improvement focus was to reduce the load time. This was done through the following approaches:

1. Fixing trivial variables, typically to zero. Some examples include; the shifts which a physicians does not have the competencies for, and the combinations of shifts and add-ons which are not allowed.

2. Pre-construct new parameters which removed unnecessary complexity and calculations from constraints. One example of this is the parameter $C^{\text{pos}}_{p,c,s,a}$, which is the consolidation of seven other parameters that further enables the reduction of the competency dimension.

3. The addition of a 'helper' decision variable, $x^{\text{help}}_{p,s,a}$, in which the day dimension is omitted.

The resultant load time was decreased to $\sim$ 5 minutes. However, we expect further improvements would be possible.

The second and main improvement focus was to increase the performance of the model either in terms of bounds or solution time. We focused on the methodologies; lexicographical optimization and decomposition. Based on discussions with our supervisor and the findings in the literature study, we found these two methodologies to be most promising. The lexicographical optimization approach consists of two stages. In the second stage only the fairness objectives are optimized, and the remaining objectives are optimized in the first stage. A parameter study was conducted in *Parameter study of the Two Stage model* to determine the best global setting for the solution gap in stage one and the slack on the first stage objective in stage two. The best combination based on the two instances considered in the analysis was a 15% optimality gap and a 15% slack. An interesting observation from the parameter study was that the full amount of slack was utilized in all cases. In *Practical quality of the Fairness model* it was shown that in one instance the 'cost' of the added fairness was $\sim 35\%$, so intuitively it makes sense the best setting in the Two Stage model allows for a similar combined slack.
It is difficult to conclude that the Two Stage model improved either the bounds or solution time as the results were conflicting in the two instance months. In July the Two Stage model outperformed the regular Fairness model in terms of both speed and bounds whereas the opposite was the case for September. It could be that the chosen Two Stage setting simply cuts away the promising solutions found by the regular Fairness model. However, we did not choose to test higher slack settings, as you eventually risk loosing control of the first stage objectives.

The Two Stage model inspired a warm-start approach on the fairness model, as the Two Stage model appears to converge fast consistently when given an initial solution from the first stage. Thus, in the new warm start approach, the initial solution from the first stage was provided to the Fairness model. In Figure 6.12 it was seen that the warm start approach had the intended effect. Once the initial solution is provided as a warm start, the Fairness model converges fast in both instances. An 15% optimally gap in the first stage was tested in the warm start approach, which took up to ~ 30 minutes to reach. To save time, it could be intriguing to test even higher optimality gaps in the first stage based on the 'cost' of fairness discussed earlier.

Lastly, a *Decomposition model* was introduced as another approach for improving bounds or solution time. The solution time of the Decomposition model was many times worse than any other approach tested. In the time study of each element in the Decomposition model, cf. Figure 6.14, it was shown that currently the load time of both the master- and subproblems accounts for the majority of the optimization time. However, even if the load time could somehow be eliminated, the combined solution time of the subproblems still exceeds the solution time of the other approaches tested.
Moreover, the Decomposition model does not even include fairness elements. Thus, the Decomposition approach requires a lot more work before it can compare to the other approaches tested.

7.3 Recommendations and applications

Based on the various analyses carried out in *Analysis and results*, we evaluate that the best solution approach is to run the *Fairness model* with a warm start. It was seen in Figure 6.12 in *Fairness model with warm start* that the Fairness model with a warm start overall performed best on the instances tested. Furthermore, in *Analysis of the optimality gap from a practically perspective* it was shown that the practical quality of the Fairness model with a warm start is good when run overnight but also almost as good when run with a two hour time limit. Conclusively why we believe this solution approach is the best tested.

Besides generating rosters, we also see strategic potential in the overall model. Every year the schedulers must orchestrate the summer holidays for their respective department. Each physician submits their desired holiday periods, from which the scheduler must evaluate if enough resources are available to fulfill all demand. This process is both time consuming and difficult for the scheduler and can require multiple iterations.
The first stage of the warm start Fairness model can alleviate both the work and time required for this process. Here, the scheduler sets up the planned demand for a holiday month along with the holiday requests, only. These requests are added as starred wishes. Since the scheduler is only interested in evaluating if the holiday requests are feasible, there is no need to run the entire Fairness model. If there is a shortage of resources with respect to demand, the solution of the first stage of the warm start Fairness model will specify for which demand(s) and on which day(s) the shortage occur. This is because the demand is implemented as soft constraints with slack, that only activates as a last resort. With a few additions to the model other strategic aspects could be evaluated. As an example the cost of over hours versus hiring new physicians can be assessed. This could be done by adding an upper limit for the total amount of over hours as well as by adding pseudo-physician.

7.4 Real life application

The *Fairness model* was used to generate the roster for the months of July, August, and September in the anesthesia department at Rigshospitalet. Each roster generated by the model was checked and adjusted by the scheduler before the roster was finalized. This process took the scheduler between 4-6 hour. The time the scheduler spend on checking and adjusting, respectively, was approximately the same. However, the scheduler expressed that the majority of the 4-6 hours was spend on copying and adjusting the format of the roster provided by the model to fit his own planning sheet. When changes are made, his own planning sheet automatically updates demand counters and the total working hours, to name a few. This makes the manual process of adding changes and adjustments more transparent for the scheduler. Developing an effective dynamic interface in Excel would be both complex and time consuming. Thus, the ideal output of the model would be a dynamic interface in a customized platform.
By comparison, the process of manually developing the roster without assistance from the model takes the scheduler 1-2 days depending on the instance.

Initially, for the month of July, the scheduler evaluated the overall quality of the roster provided by the model to be $\sim 80\%$ of his definition of optimality. The most common adjustments made by the scheduler was trying to grant more wishes and fix general inconveniences; sometimes at the expense of violating regulations. By fixing errors in the data set, making minor adjustments to the model, and encouraging the scheduler to star-mark more working wishes, the scheduler evaluated the quality of the September plan to be close to $\sim 90\%$ from optimum.

7.5 Modeling aspects

From our perspective, the most challenging aspect of developing a model which could generate a monthly roster for the anesthesia department, was the inherent complexity of the problem. The set of regulations which governs the work environment for physicians grows increasingly complicated with the constant addition of new rules. From our understanding, the various stakeholders in the work environment of physicians agree that

the current set of regulations are too complex, but it is too extensive of a task to reform them. This complexity has a big influence on the setup and daily running of the anesthesia department.
The complexity is apparent through the large amount of data presented in the *Data introduction* section. There are various data elements and the dependencies between them further complicates the data structure. To accommodate for all the elements of the problem, the resulting model is also large and complex. The full MIP Fairness model contains 101 abstracts constraints and an objective which minimizes 22 abstract elements. The advantage of modeling complex problems as MIP models, is that the modeling process is relatively straight forward as the complexity is broken down in sets of constraints.

MIP models utilize an exact solution approach where the optimality of any solution can be evaluated by the solution bounds. It can be argued that the physician rostering problem is heuristic by nature, as true optimality cannot be defined. This is due to the various humanistic elements of the problem. Therefore, e.g. metaheuristic would be interesting to consider as well. However, there are multiple reason why we did not deem metaheuristics to be a promising research direction for our problem. Firstly, given the size and complexity of the problem, implementing even a local search algorithm is difficult because many illegal moves exist. Implementing a good construction heuristic is also not a simple task. Lastly, maintaining a metaheuristic is also significantly more difficult compared to a MIP model, because the addition of new elements could require complex changes to the algorithm.

7.5.1 Multi objectivity and scalability

As already mentioned in the *Literature study* the weighted sum approach is not an ideal way to handle a mix of financial and non-financial objectives. However, due to a weighted sum approach being a simple way to construct an objective function, we chose to implement this approach given the time horizon of this thesis. Through the tuning process we found that the value of the weighted coefficients had an influence on both the solution and the solution time. However, within the scope of this thesis we tuned the coefficient ad hoc based on the inputs from the anesthesia scheduler, in order to achieve a good practical quality. This unexplored part of the model is promising in terms of potentially improving the solution quality and time further.

We would also like to stress that the scalability of the developed model is questionable. Through the literature study we came across elements of the physician scheduling problem which were not relevant at the antithesis department, but crucial at other hospitals. E.g. the possibility to wish for whom to work with on a shift or the ability to pre-schedule breaks as part of the roster.

7.6 Further work

As the last part of the discussion, relevant further work is detailed. The purpose of this section is to outline the next phases from an academic and a commercial perspective with the common goal of strengthen the performance and application of the model.

7.6.1 Academic perspective

Although different optimization techniques have been explored, the optimality gap is still high from a theoretical perspective, over 8% in all cases and in most cases over 15%, cf. *Analysis and results*. Therefore, it is still interesting to research how to optimize the performance of the model further. A key element that should be implemented is a preprocessing step of the data. As an example the 'add-on' dimension can be merged with the 'shift' dimension which would reduce the loading time of the model by a factor of approximately 6 (the number of add-ons).

An interesting methodology to investigate to speed up the optimization process of the model is the use of cutting plane generation. The idea of cutting plane generation is to cut off feasible solutions to the LP-relaxation by solving the separation problem, or to simply cut away unwanted solutions decreasing the overall solution space. Lodi [2013] and Böðvarsdóttir et al. [2021] both argue that the two methods look promising in terms of improving the performance of a model for a nurse rostering problem. Multiple optimization techniques can also be merged, with the purpose of decreasing the optimization time further and obtain better rosters.

Another important area to further investigate is the entire multi-objectivity perspective. In this thesis a simple deterministic weighted sum approach is utilised, even though it has received criticism [Böðvarsdóttir et al., 2021]. Therefore, it could be interesting to see if other approaches could improve the quality further. A different way of handling the multi-objectivity is utilising Goal Programming. Goal Programming is a methodology where specific targets/goals are created. This approach is in general one of the easier methods to implement, however choosing the best fitting variant and appropriate goals is a complex procedure [Jones and Tamiz, 2010]. Another intriguing approach for tackling the multi-objectivity is Behind-the-Scenes Weight Tuning. This approach uses measurable targets for guidance in order to automatically set weights, i.e. schedulers are not required to provide accurate objective weights themselves, resulting in substantially reduced manual effort and better rosters as compared to a weighted sum approach [Böðvarsdóttir et al., 2021].

Lastly, another interesting area to study further is the consistency of the performance of the model with respect to time and quality. In this study three schedules were generated for the anesthesia department. From a practical viewpoint using the generated schedules saved the scheduler time. However, from a theoretical viewpoint the optimality varied between 8% to 15% and the optimization process depended a lot on the individual instance. Thus, further instances should be studies in order to ensure that the quality of the model is sufficient in all cases. Furthermore, it will provide a better basis for validating optimization techniques.

7.6.2 Commercial perspective

During this thesis it has been demonstrated that the quality of the schedules and the time it takes to produce them is sufficient from a practical perspective. Thus, key elements to consider from a commercial perspective is the entire development of a proper data structure and a Graphic User Interface (GUI). A proper data structure and interface would ease the process of scaling the model both in internally and to other departments. The GUI is a key element to advance the model from an academic project to a commercial product. Furthermore, the GUI would ideally save the schedulers time as it would be more intuitive and efficient to update and maintain data, and to make changes to schedules.

Another crucial thing to consider is to expand the study to more departments to test the scaleability across different set-ups and to ensure that all key elements are included in the model. With a proper data structure and a GUI, it will be easier to include and set up more departments. Moreover, the scheduler from each department can be included in the process, hence making the transition for the scheduler more smooth.

8 Conclusion

The aim of this thesis was to develop a model that could provide decision support for the process of generating a monthly roster for the anesthesia department at Rigshospitalet. The model should minimize the manual efforts needed from the scheduler while maintaining a certain level of generality for the sake of potentially scaling the model to other departments or hospitals.

The thesis has been realized in collaboration with the Anesthesia department and the Unit for Data and Quality at Rigshospitalet. The Anesthesia department has kindly provided the insight and data needed to construct a model which could generate real rosters. The rosters subsequently used by the anesthesia department in June, July, and August were the product of: the schedules provided by the proposed model in this thesis and final modifications made by the scheduler.

The chosen model approach was a Mixed Integer Programming (MIP) model with a weighted sum objective function that minimizes a set of both financial and non-financial abstract objectives. To enable the development of the MIP model the underlying data was initially compiled and structured into a set of 13 data sheets. The output of the developed model, as described in *Practical quality of the Fairness model*, is a set of 11 data sheets providing the scheduler with, first of all, the proposed roster for a given month along with various overviews and performance indicators.
The development process of the model was split into three stages: a *Base model* which includes the most essential scheduling requirements, a *MVP model* which further includes all relevant requirements and features except fairness elements, and a *Fairness model* which includes the same requirements and features as the MVP model as well as fairness elements. The Fairness model is the final product of the three stage development process.

The Fairness model was not able to obtain solutions with a $< 10\%$ optimality gap in less than 24 hours. Moreover, the loading time of model was significant and initially took close to two hours. Methods to improve both the loading time and the optimization process was investigated and discussed in *Analysis and results* and *Solution methods*, respectively. By fixing trivial variables, pre-constructing new parameters, and adding a 'helper' decision variable, the load time of the Fairness model was reduces to ~ 5 minutes.
Three different solution approaches were explored in the search of improving the optimization process of the Fairness model: a *Decomposition model*, a *Two stage model*, and a warm start approach on the Fairness model. These three approaches were evaluated based on two problem instances: July and September. The decomposition approach did not improve the optimization process by either bound nor time. Both the loading time of the sub- and master-problem(s) as well as the solution time of the subproblems were found to be too time consuming for the approach to have relevance within the time-frame of this project. The quality of the solutions found by the Two Stage model were inconsistent. In July the Two Stage model outperformed the regular Fairness model whereas in September it was the other way around. Nevertheless, the consistent convergence trend seen in the second stage of the Two Stage model in both months seemed promising and thus inspired a warm start approach. In the instances considered, the Fairness model with a warm start converged more consistently than without a warm start, without compromising on solution quality. Based on the analyses carried out, the Fairness model with a warm start had the best overall performance.

The optimality gap obtained after running the Fairness model with a warm start for 12 hours was 18% for July and 15% for September. With a run-time of two hours the optimality gaps obtained were 23% and 28%, respectively. Despite the obtained solutions being far from theoretical optimality, the resultant practical quality was valuable. After a two hour run, few unnecessary outliers can still occur, whereas after 12 hours only necessary outliers seem to remain. Generally, the KPIs suggest that the model works as intended. Demand and targets are fulfilled, granted wishes appears to be maximized and so on. The fairness elements seem to indeed promote fairness and decrease the variance for the respective KPIs.

The scheduler in the anesthesia department evaluated the quality of the July plan to be $\sim$ 80% of his definition of optimality. After minor adjustments to the data and the model based on feedback from the scheduler, the scheduler evaluated the quality of the subsequent September plan to be near 90% from optimum. The time spend on manually creating a roster without the assistance of the model is between 1-2 work days. Using the roster generated by the model this time is reduced to 4-6 hour. The majority of this time the scheduler spends on converting the format of the generated roster to his own dynamic sheet, which he prefers to use for making adjustments.

Based on the various analyses carried out and the feedback from the scheduler, we conclude that the aim of generating rosters which minimize the adjustments and efforts needed by the scheduler, has been achieved to a satisfactory extend within the scope of this thesis.

Bibliography

Abdalkareem, Z. A., Amir, A., Al-Betar, M. A., Ekhan, P., and Hammouri, A. I. (2021). Healthcare scheduling in optimization context: a review. *Health and Technology*, 11:445–469.

Bixby, R. and Rothberg, E. (2007). Progress in computational mixed integer programming - a look back from the other side of the tipping point. *Annals of Operations Research*, 149:37–41.

Branke, J., Deb, K., Miettinen, K., and Słowiński, R. (2008). *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer-Verlag, Berlin, Heidelberg.

Brunner, J. O., Bard, J. F., and Kolisch, R. (2009). Flexible shift scheduling of physicians. *Health Care Management Science*, 12:285–305.

Burke, E., De Causmaecker, P., Berghe, G., and Van Lendeghem, H. . (2004). The state of the art of nurse rostering. *Journal of Scheduling*, 7:441–499.

Böðvarsdóttir, E. B., Smet, P., Van den Berghe, G., and Stidsen, T. J. (2021). Achieving compromise solutions in nurse rostering by using automatically estimated acceptance thresholds. *European Journal of Operational Research*, 292(3):980–995.

Cayirli, T. and Veral, E. A. (2009). Modeling and solving the dynamic patient admission scheduling problem under uncertainty. *Production and Operations Management*, 12(4):519–549.

Ceschia, S. and Schaerf, A. (2012). Modeling and solving the dynamic patient admission scheduling problem under uncertainty. *Artificial Intelligence in Medicine*, 56(3):199–205.

Cheang, B., Li, H., Lim, A., and Rodrigues, B. (2003). Nurse rostering problems - a bibliographic survey. *European Journal of Operational Research*, 151(3):447–460.

Cildoz, M., Mallor, F., and Mateo, P. (2021). A grasp-based algorithm for solving the emergency room physician scheduling problem. *Applied Soft Computing*, 219:622–629.

Clarke, M. D. D. and Ryan, D. M. (2013). Airline industry operations research. In *Encyclopedia of Operations Research and Management Science*, pages 37–43. Springer US, Boston, MA.

Dantzig, G. B. and Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research*, 8(1):101–111.

de Bruin, A. M., Koole, G., and Visser, M. C. (2006). Bottleneck analysis of emergency cardiac in-patient flow in a university setting: An application of queueing theory. *Clinical and investigative medicine*, 28(6):316–7.

de Kreuk, A. C. C., Winands, E. M. M., and Vissers, J. (2004). Master scheduling of medical specialists. In *Master scheduling of medical specialists*, pages 184–201. NY Routledge, New York, USA.

Demeester, P., Souffriau, W., De Causmaecker, P., and Van den Berghe, G. (2010). A hybrid tabu search algorithm for automatically assigning patients to beds. *Artificial Intelligence in Medicine*, 48(1):61–70.

El-Rifai, O., Garaix, T., Augusto, V., and Xie, X. (2015). A stochastic optimization model for shift scheduling in emergency departments. *Health Care Management Science*, 18(3):289–302.

Erhard, M., Schoenfelder, J., Fügener, A., and Brunner, J. O. (2018). State of the art in physician scheduling. *European Journal of Operational Research*, 265:1–18.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B., and Sier, D. (2004a). An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, 127(1-4):21 – 144.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004b). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27. Timetabling and Rostering.

Green, L. V., Soares, J., Giglio, J. F., and Green, R. A. (2006). Using queueing theory to increase the effectiveness of emergency department provider staffing. *Society for Academic Emergency Medicine*, 13(1):61–68.

Guerriero, F. and Guido, R. (2011). Operational research in the management of the operating theatre: a survey. *Health Care Management Science*, 14:89–114.

Jones, D. and Tamiz, M. (2010). *Practical Goal Programming*. Springer US, New York, USA.

Lamirim, M., Grimaud, F., and Xie, X. (2009). Optimization methods for a stochastic surgery planning problem. *International Journal of Production Economics*, 120(2):400–410.

Lau, H. C. (1996). On the complexity of manpower shift scheduling. *Computers & Operations Research*, 23(1):93–102.

Lavoie, S., Minoux, M., and Odier, E. (1988). A new approach for crew pairing problems by column generation with an application to air transportation. *European Journal of Operational Research*, 35(1):45–58.

Lodi, A. (2013). The heuristic (dark) side of mip solvers. In Talbi, E.-G., editor, *Hybrid Metaheuristics*, pages 273–284. Springer Berlin Heidelberg, Berlin, Heidelberg.

Min, D. and Yih, Y. (2010). Scheduling elective surgery under uncertainty and downstream capacity constraints. *European Journal of Operational Research*, 206:642–652.

Petrovic, S. (2019). " you have to get wet to learn how to swim" applied to bridging the gap between research into personnel scheduling and its implementation in practice. *Annals of Operations Research*, 275:161–179.

Queyranne, M. and Wolsey, L. A. (2017). Tight mip formulations for bounded up-/down times and interval-dependent start-ups. *Mathematical Optimization Society*, 164:129–155.

Rais, A. and Viana, A. (2011). Operations research in healthcare: a survey. *International Transactions in Operational Research*, 18(1):1–31.

Rousseau, L. M., Gendreau, M., and Peasant, G. (2002). A general approach to the physician rostering problems. *Annals of Operations Research*, 115:193–205.

Saadouli, H., Jerbi, B., Dammak, A., Masmoudi, L., and Bouaziz, A. (2015). A stochastic optimization and simulation approach for scheduling operating rooms and recovery beds in an orthopedic surgery department. *Computers & Industrial Engineering*, 80:72–79.

Topaloglu, S. and Ozkarahan, I. (2011). A constraint programming-based solution approach for medical resident scheduling problems. *Computers Operations Research*, 38(1):246–255.

Ullman, J. (1975). Np-complete scheduling problems. *Journal of Computer and System Sciences*, 10(3):384–393.

Vanberkel, P. T., Boucherie, R. J., Hans, E. W., Hurink, J. L., and Litvak, N. (2009). A survey of health care models that encompass multiple departments. *International Journal of Health Management and Information*, 1(1):37–69.

White, C. A. and White, G. M. (2003). Scheduling doctors for clinical training unit rounds using tabu optimization. In Burke, E. and De Causmaecker, P., editors, *Practice and Theory of Automated Timetabling IV*, pages 120–128, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yngre Læger (2017). Arbejdstid. https://www.laeger.dk/arbejdstid-yngre-laeger. Last accessed on Jun 24, 2021.

A Complete Fairness model

This appendix presents an overview of all of the constraints and the objective that make up the complete *Fairness model*. This includes the objective, (O3), from the *Fairness model*, all of the constraints, (C1.1)-(C1.21), from the *Base model*, all of the constraints, (C2.1)-(C2.73), from the *MVP model*, and all of the constraints, (C3.1)-(C3.10), from the *Fairness model*:

$$
\begin{aligned}
\text{Min} \quad & \sum_{p\in\mathcal{P}} (St_p^{\text{oHr}}\, h_p^{\text{ov}} + 10\, h_p^{\text{un}}) + \sum_{p\in\mathcal{P}} (30\, St_p^{\text{oHr}}\, NR_p\, h_p^{\text{ov}}) + \\
& \sum_{p\in\mathcal{P}} \sum_{c\in\mathcal{C}^{\text{perc}}} (hC_{p,c}^{\text{ov}} + 5\, hC_{p,c}^{\text{un}}) + \sum_{c\in\mathcal{C}^{\text{prio}}} p_c^{\text{prio}} + 10 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} p_{p,d}^{\text{wish}} + \\
& 10 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} p_{p,d}^{\text{adm}} + 10 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} \sum_{s\in\mathcal{S}} p_{p,d,s}^{\text{admS}} + 10 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} \sum_{a\in\mathcal{A}} p_{p,d,a}^{\text{admA}} + \\
& 50 \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} pWkd_{p,q}^{\text{work}} + 100 \sum_{p\in\mathcal{P}} pFD_p^{\text{u}} + 1000\, p^{\text{OLG}} + \\
& 500 \sum_{c\in\mathcal{C}} \sum_{d\in\mathcal{D}} \left(o_{c,d}^{\text{dem}} + u_{c,d}^{\text{dem}}\right) + 5 \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} p_{p,d}^{\text{row}} + \\
& 20 \sum_{p\in\mathcal{P}} \left(p_p^{\text{pres}} + p_p^{\text{alert}} + p_p^{\text{call}}\right) + 10 \sum_{p\in\mathcal{P}} \sum_{d\in D^{\text{all}}} p_{p,d}^{\text{YL}} + \\
& \sum_{p\in\mathcal{P}} \left(50\, pF_{p,1}^{\text{w}} + 200\, pF_{p,2}^{\text{w}} + 800\, pF_{p,3}^{\text{w}}\right) + \\
& \sum_{p\in\mathcal{P}} \left(200\, pF_{p,1}^{\text{oH}} + 800\, pF_{p,2}^{\text{oH}} + 2600\, pF_{p,3}^{\text{oH}}\right) + \\
& \sum_{p\in\mathcal{P}} \sum_{c\in\mathcal{C}^{\text{f}}} \left(1000\, pF_{p,c,1}^{\text{cou}} + 2000\, pF_{p,c,2}^{\text{cou}} + 3000\, pF_{p,c,3}^{\text{cou}}\right)
\end{aligned}
\tag{O7}
$$

Subject to

$$\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} St_p^{\text{emp}}\, x_{p,d,s,a} = 0, \qquad \forall p\in\mathcal{P},\ \forall d\in\mathcal{D} \tag{C7.1}$$

$$\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} x_{p,d,s,a} = 1, \qquad \forall p\in\mathcal{P},\ \forall d\in\mathcal{D};\ St_p^{\text{emp}} = 0 \tag{C7.2}$$

$$\sum_{a\in\mathcal{A}} St_p^{\text{shi}}\, x_{p,d,s,a} = 0, \qquad \forall p\in\mathcal{P},\ \forall d\in\mathcal{D},\ \forall s\in\mathcal{S} \tag{C7.3}$$

$$\sum_{d\in\mathcal{D}} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} x_{p,d,s,a}\, (H_{p,s}^{\text{frac}}\ H_s^{\text{shi}} + H_a^{\text{add}}) - H_p^{\text{targ}} \le h_p^{\text{ov}}, \qquad \forall p\in\mathcal{P} \tag{C7.4}$$

$$H_p^{\text{targ}} - \sum_{d\in\mathcal{D}} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} x_{p,d,s,a} (H_{p,s}^{\text{frac}}\ H_s^{\text{shi}} + H_a^{\text{add}}) \le h_p^{\text{un}}, \qquad \forall p\in\mathcal{P} \tag{C7.5}$$

$$\sum_{a\in\mathcal{A}} x_{p,d,s,a} \le D_{s,v}^{\text{shi}}, \qquad \forall p\in\mathcal{P},\ \forall d\in\mathcal{D},\ \forall s\in\mathcal{S},\ v = \{D_d^{\text{num}}\} \tag{C7.6}$$

$$K_{p,k}^{\text{emp}} \ge x_{p,d,s,a}\ K_{s,k}^{\text{shi}}, \qquad \forall p\in\mathcal{P},\ \forall d\in\mathcal{D},\ \forall s\in\mathcal{S},\ \forall a\in\mathcal{A},\ \forall k\in\mathcal{K} \tag{C7.7}$$

$$K^{\text{emp}}_{p,k} \geq x_{p,d,s,a}\ K^{\text{add}}_{a,k}, \qquad \forall p \in \mathcal{P},\, \forall d \in \mathcal{D},\, \forall s \in \mathcal{S},\, \forall a \in \mathcal{A},\, \forall k \in \mathcal{K} \tag{C7.8}$$

$$x_{p,d,s,a} \leq\ SA^{\text{com}}_{s,a}, \qquad \forall p \in \mathcal{P},\, \forall d \in \mathcal{D},\, \forall s \in \mathcal{S},\, \forall a \in \mathcal{A} \tag{C7.9}$$

$$\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C^{\text{pos}}_{p,c,s,a}\ x_{p,d,s,a} = C^{\text{dem}}_{c,d} + o^{\text{dem}}_{c,d} - u^{\text{dem}}_{c,d} \qquad \forall d \in \mathcal{D}, \forall c \in \mathcal{C},\, C^{\text{type}}_{c} = \{P\} \tag{C7.10}$$

$$\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C^{\text{pos}}_{p,c,s,a}\ x_{p,d,s,a} \geq C^{\text{dem}}_{c,d} - u^{\text{dem}}_{c,d} \qquad \forall d \in \mathcal{D}, \forall c \in \mathcal{C},\, C^{\text{type}}_{c} = \{M\} \tag{C7.11}$$

$$\sum_{d\in\mathcal{D}} x_{p,d,s,a} = x^{\text{help}}_{p,s,a}, \qquad \forall p \in \mathcal{P},\, \forall s \in \mathcal{S},\, \forall a \in \mathcal{A} \tag{C7.12}$$

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C^{\text{shi}}_{c,s}\ C^{\text{add}}_{c,a}\ x^{\text{help}}_{p,s,a} \geq C^{\text{days}}_{p,c}, \qquad \forall p \in \mathcal{P}, \forall c \in \mathcal{C}^{\text{days}} \tag{C7.13}$$

$$\left(H^{\text{targ}}_{p} - \sum_{c_1\in\mathcal{C}^{\text{days}}} C^{\text{dem}}_{c_1,d}\, \frac{\sum_{s_1\in\mathcal{S}} C^{\text{shi}}_{c_1,s_1}\ H^{\text{frac}}_{p,s_1}\, H^{\text{shi}}_{s_1}}{\sum_{s_2\in\mathcal{S}} C^{\text{shi}}_{c_1,s_2}} - \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x^{\text{help}}_{p,s,a}\ S^{\text{oth}}_{s}\ H^{\text{frac}}_{p,s}\ H^{\text{shi}}_{s} \right) \frac{C^{\text{perc}}_{p,c}}{100}$$
$$\leq \sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \left(x_{p,d,s,a}\, C^{\text{shi}}_{c,s}\ H^{\text{frac}}_{p,s}\ H^{\text{shi}}_{s} \right) + hC^{\text{un}}_{p,c}, \qquad \forall p \in \mathcal{P}, \forall c \in \mathcal{C}^{\text{perc}} \tag{C7.14}$$

$$\left(H^{\text{targ}}_{p} - \sum_{c_1\in\mathcal{C}^{\text{days}}} C^{\text{dem}}_{c_1,d}\, \frac{\sum_{s_1\in\mathcal{S}} C^{\text{shi}}_{c_1,s_1}\ H^{\text{frac}}_{p,s_1}\, H^{\text{shi}}_{s_1}}{\sum_{s_2\in\mathcal{S}} C^{\text{shi}}_{c_1,s_2}} - \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x^{\text{help}}_{p,s,a}\ S^{\text{oth}}_{s}\ H^{\text{frac}}_{p,s}\ H^{\text{shi}}_{s} \right) \frac{C^{\text{perc}}_{p,c}}{100}$$
$$\geq \sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \left(x_{p,d,s,a}\, C^{\text{shi}}_{c,s}\ H^{\text{frac}}_{p,s}\ H^{\text{shi}}_{s} \right) - hC^{\text{ov}}_{p,c}, \qquad \forall p \in \mathcal{P}, \forall c \in \mathcal{C}^{\text{perc}} \tag{C7.15}$$

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,d,s,a}\, WS^{\text{add}}_{w,a}\, WS^{\text{shi}}_{w,s} = 1, \qquad \forall p \in \mathcal{P},\, \forall d \in \mathcal{D},\, w = \{WP^{\text{shi}}_{p,d}\};\, WP^{\text{prio}}_{p,d} = \{*\} \wedge WP^{\text{shi}}_{p,d} \neq \emptyset \wedge WP^{\text{add}}_{p,d} = \emptyset \tag{C7.16}$$

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,d,s,a}\, WA^{\text{add}}_{w,a}\, WA^{\text{shi}}_{w,s} = 1, \qquad \forall p \in \mathcal{P},\, \forall d \in \mathcal{D},\, w = \{WP^{\text{add}}_{p,d}\};\, WP^{\text{prio}}_{p,d} = \{*\} \wedge WP^{\text{shi}}_{p,d} = \emptyset \wedge WP^{\text{add}}_{p,d} \neq \emptyset \tag{C7.17}$$

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,d,s,a}\, WS^{\text{add}}_{w_1,a}\, WS^{\text{shi}}_{w_1,s}\, WA^{\text{add}}_{w_2,a}\, WA^{\text{shi}}_{w_2,s} = 1, \qquad \forall p \in \mathcal{P},\, \forall d \in \mathcal{D},\, w_1 = \{WP^{\text{shi}}_{p,d}\},\ w_2 = \{WP^{\text{add}}_{p,d}\};\, WP^{\text{prio}}_{p,d} = \{*\} \wedge WP^{\text{shi}}_{p,d} \neq \emptyset \wedge WP^{\text{add}}_{p,d} \neq \emptyset \tag{C7.18}$$

$$\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C^{\text{pos}}_{p,c,s,a}\ x^{\text{help}}_{p,s,a}\ PP^{\text{shi}}_{c,s} \leq p^{\text{prio}}_{c}, \qquad \forall c \in \mathcal{C}^{\text{prio}};\ C^{\text{prio}}_{c} = \{\text{Shift}\} \tag{C7.19}$$

$$\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C^{\text{pos}}_{p,c,s,a}\ x^{\text{help}}_{p,s,a}\ PP^{\text{add}}_{c,a} \leq p^{\text{prio}}_{c}, \qquad \forall c \in \mathcal{C}^{\text{prio}};\ C^{\text{prio}}_{c} = \{\text{Add-on}\} \tag{C7.20}$$

$$\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C^{\text{pos}}_{p,c,s,a}\ x^{\text{help}}_{p,s,a} \left(1 - \sum_{k\in\mathcal{K}} K^{\text{emp}}_{p,k}\ BP^{\text{comp}}_{c,k} \right) \leq p^{\text{prio}}_{c}, \qquad \forall c \in \mathcal{C}^{\text{prio}};\ C^{\text{prio}}_{c} = \{\text{Competency}\} \tag{C7.21}$$

$$z_{p,d,s,a} = x_{p,d,s,a}, \qquad \forall p \in \mathcal{P}, \forall d \in \mathcal{D}, \forall s \in \mathcal{S}, \forall a \in \mathcal{A} \tag{C7.22}$$

$$z_{p,d,s,a} = 1 \qquad \forall p \in \mathcal{P},\ \forall d \in \mathcal{D}^{\text{prev}},\ s = \{OP^{\text{shi}}_{p,d}\},\ a = \{OP^{\text{add}}_{p,d}\} \tag{C7.23}$$

$$z_{p,d,s,a} = 0 \qquad \forall p \in \mathcal{P},\ \forall d \in \mathcal{D}^{\text{prev}},\ s = \{OP^{\text{shi}}_{p,d}\},\ \forall a \in \mathcal{A}\backslash\{OP^{\text{add}}_{p,d}\} \tag{C7.24}$$

$$z_{p,d,s,a} = 0 \qquad \forall p \in \mathcal{P},\ \forall d \in \mathcal{D}^{\text{prev}},\ \forall s \in \mathcal{S}\backslash\{OP^{\text{shi}}_{p,d}\},\ \forall a \in \mathcal{A} \tag{C7.25}$$

$$z_{p,d,s,a} = 1 \qquad \forall p \in \mathcal{P},\ \forall d \in [d^{e+1}, ..., d^{e+4}],\ s = \{\text{FRI\_h}\},\ a = \{\text{ingen}\} \tag{C7.26}$$

$$z_{p,d,s,a} = 0 \qquad \forall p \in \mathcal{P},\ \forall d \in [d^{e+1}, ..., d^{e+4}],\ s = \{\text{FRI\_h}\},\ \forall a \in \mathcal{A}\backslash\{\text{ingen}\} \tag{C7.27}$$

$$z_{p,d,s,a} = 0 \qquad \forall p \in \mathcal{P},\ \forall d \in [d^{e+1}, ..., d^{e+4}],\ \forall s \in \mathcal{S}\backslash\{\text{FRI\_h}\},\ \forall a \in \mathcal{A} \tag{C7.28}$$

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} x_{p,d,s,a}\, WS^{\text{add}}_{w,a}\, WS^{\text{shi}}_{w,s} + p^{\text{wish}}_{p,d} = 1, \qquad \forall p \in \mathcal{P},\ \forall d \in \mathcal{D},\ w = \{WP^{\text{shi}}_{p,d}\};\ WP^{\text{prio}}_{p,d} \neq \{*\} \wedge WP^{\text{shi}}_{p,d} \neq \emptyset \wedge WP^{\text{add}}_{p,d} = \emptyset \tag{C7.29}$$

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} x_{p,d,s,a}\, WA^{\text{add}}_{w,a}\, WA^{\text{shi}}_{w,s} + p^{\text{wish}}_{p,d} = 1, \qquad \forall p \in \mathcal{P},\ \forall d \in \mathcal{D},\ w = \{WP^{\text{add}}_{p,d}\};\ WP^{\text{prio}}_{p,d} \neq \{*\} \wedge WP^{\text{shi}}_{p,d} = \emptyset \wedge WP^{\text{add}}_{p,d} \neq \emptyset \tag{C7.30}$$

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} x_{p,d,s,a}\, WS^{\text{add}}_{w_1,a}\, WS^{\text{shi}}_{w_1,s}\, WA^{\text{add}}_{w_2,a}\, WA^{\text{shi}}_{w_2,s} + p^{\text{wish}}_{p,d} = 1, \qquad \forall p \in \mathcal{P},\ \forall d \in \mathcal{D},\ w_1 = \{WP^{\text{shi}}_{p,d}\},\ w_2 = \{WP^{\text{add}}_{p,d}\};\ WP^{\text{prio}}_{p,d} \neq \{*\} \wedge WP^{\text{shi}}_{p,d} \neq \emptyset \wedge WP^{\text{add}}_{p,d} \neq \emptyset \tag{C7.31}$$

$$\sum_{a \in \mathcal{A}} z_{p,d,s,a}\, RS^{\text{pres}}_{s} \leq \sum_{a_1 \in \mathcal{A}} z_{p,d+1,s_1,a_1} \qquad \forall p \in \mathcal{P}, \forall d \in [d^{-1}, .., d^{e-1}], \forall s \in \mathcal{S}, s_1 = \{SOV\},\ k = \{NAT\};\ K^{\text{shi}}_{s,k} = 1 \tag{C7.32}$$

$$\sum_{s \in \mathcal{S}} z_{p,d,s,a}\, RA^{\text{pres}}_{a} \leq \sum_{a_1 \in \mathcal{A}} z_{p,d+1,s_1,a_1} \qquad \forall p \in \mathcal{P},\ \forall d \in [d^{-1}, ..., d^{e-1}],\ s_1 = \{SOV\},\ \forall a \in \mathcal{A},\ k = \{NAT\};\ K^{\text{add}}_{a,k} = 1 \tag{C7.33}$$

$$\sum_{a \in \mathcal{A}} z_{p,d,s,a} \leq \sum_{a_1 \in \mathcal{A}} z_{p,d+1,s_1,a_1} + \sum_{s_2 \in \mathcal{S}} \left((R^{\text{pref}}_{p} + R^{\text{can}}_{p})\, S^{\text{adm}}_{s_2} \sum_{a_2 \in \mathcal{A}} z_{p,d+1,s_2,a_2} \right) \qquad \forall p \in \mathcal{P},\ \forall s \in \mathcal{S},\ \forall d \in [d^{-1}, .., d^{e-1}],\ s_1 = \{SOV\},\ k = \{NAT\};\ K^{\text{shi}}_{s,k} = 1 \tag{C7.34}$$

$$\sum_{s\in\mathcal{S}} z_{p,d,s,a} \leq \sum_{a_1\in\mathcal{A}} z_{p,d+1,s_1,a_1} + \sum_{s_2\in\mathcal{S}} \left((R^{\text{pref}}_p + R^{\text{can}}_p)\, S^{\text{adm}}_{s_2} \sum_{a_2\in\mathcal{A}} z_{p,d+1,s_2,a_2} \right)$$
$$\forall p \in \mathcal{P},\, \forall a \in \mathcal{A},\, \forall d \in [d^{-1}, .., d^{e-1}],\, s_1 = \{SOV\},\, k = \{NAT\};\, K^{\text{add}}_{a,k} = 1 \tag{C7.35}$$

$$\sum_{a\in\mathcal{A}} z_{p,d,s,a}\, R^{\text{can}}_p \leq \sum_{a_1\in\mathcal{A}} z_{p,d+1,s_1,a_1} + p^{\text{admS}}_{p,d+1}$$
$$\forall p \in \mathcal{P}, \forall s \in \mathcal{S}, \forall d \in [d^{-1}, .., d^{e-1}],\, s_1 = \{SOV\},\, k = \{NAT\};\, K^{\text{shi}}_{s,k} = 1 \tag{C7.36}$$

$$\sum_{s\in\mathcal{S}} z_{p,d,s,a}\, R^{\text{can}}_p \leq \sum_{a_1\in\mathcal{A}} z_{p,d+1,s_1,a_1} + p^{\text{admA}}_{p,d+1}$$
$$\forall p \in \mathcal{P}, \forall a \in \mathcal{A}, \forall d \in [d^{-1}, .., d^{e-1}],\, s_1 = \{SOV\},\, k = \{NAT\};\, K^{\text{add}}_{a,k} = 1 \tag{C7.37}$$

$$\sum_{a\in\mathcal{A}} z_{p,d+1,s,a}\, R^{\text{pref}}_p \leq p^{\text{adm}}_{p,d+1} + \sum_{s_1\in\mathcal{S}} \sum_{a_1\in\mathcal{A}} z_{p,d,s_1,a_1} (RS^{\text{pres}}_{s_1} + RA^{\text{pres}}_{a_1})$$
$$\forall p \in \mathcal{P}, \forall d \in [d^{-1}, .., d^{e-1}],\, s = \{SOV\} \tag{C7.38}$$

$$\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} x^{\text{help}}_{p,s,a} (RS^{\text{pres}}_s + RA^{\text{pres}}_a) \leq \left\lfloor \frac{D^{\text{mo}}}{R^{\text{pres}}_p} \right\rfloor + p^{\text{pres}}_p, \qquad \forall p \in \mathcal{P} \tag{C7.39}$$

$$\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} x^{\text{help}}_{p,s,a} (RS^{\text{alert}}_s + RA^{\text{alert}}_a) \leq \left\lfloor \frac{D^{\text{mo}}}{R^{\text{alert}}_p} \right\rfloor + p^{\text{alert}}_p, \qquad \forall p \in \mathcal{P} \tag{C7.40}$$

$$\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} x^{\text{help}}_{p,s,a} (RS^{\text{call}}_s + RA^{\text{call}}_a) \leq \left\lfloor \frac{D^{\text{mo}}}{R^{\text{call}}_p} \right\rfloor + p^{\text{call}}_p, \qquad \forall p \in \mathcal{P} \tag{C7.41}$$

$$t^{\text{shi}}_{p,d} \leq \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \left(H^{\text{endS}}_{s,a}\, z_{p,d,s,a} + H^{\text{startS}}_s\, z_{p,d_1,s,a} \right) +$$
$$24(d_1 - d - 1) + 24 \cdot 4 \sum_{s_1\in\mathcal{S}} \sum_{a_1\in\mathcal{A}} \left(1 - S^{\text{work}}_{s_1}\right) z_{p,d_1,s_1,a_1} \tag{C7.42}$$
$$\forall p \in \mathcal{P},\, d \in [d^{-R^{\text{wRow}}_p}, ..., d^e],\, d_1 \in [d+1, ..., d+3]$$

$$t^{\text{shi}}_{p,d} \geq \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \left(H^{\text{endS}}_{s,a}\, z_{p,d,s,a} + H^{\text{startS}}_s\, z_{p,d_1,s,a} \right) +$$
$$24\,(d_1 - d - 1) - 24 \cdot 4 \sum_{s_1\in\mathcal{S}} \sum_{a_1\in\mathcal{A}} \left(1 - S^{\text{work}}_{s_1}\right) z_{p,d_1,s_1,a_1} +$$
$$24 \cdot 4 \left(1 - \sum_{d_2=d+1}^{d_1} \sum_{s_2\in\mathcal{S}} \sum_{a_2\in\mathcal{A}} S^{\text{work}}_{s_2}\, z_{p,d_1,s_2,a_2} - \sum_{s_3\in\mathcal{S}} \sum_{a_3\in\mathcal{A}} (1 - S^{\text{work}}_{s_3})\, z_{p,d_2,s_3,a_3} \right)$$
$$\forall p \in \mathcal{P},\, d \in [d^{-R^{\text{wRow}}_p}, ..., d^e],\, d_1 \in [d+1, ..., d+3] \tag{C7.43}$$

$$t^{\text{shi}}_{p,d} = H^{\text{fDS}}_p \qquad \forall p \in \mathcal{P},\, d \in [d^s, ..., d^{-R^{\text{wRow}}_p - 1}] \tag{C7.44}$$

$$t^{\text{shi}}_{p,d} \geq H^{\text{rest}}_p - 2H^{\text{rest}}_p \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \Big(z_{p,d,s,a} \left(R^{pref}_p + R^{can}_p \right) \Big(RA^{\text{alert}}_a + RA^{\text{call}}_a + RS^{\text{alert}}_s + RS^{\text{call}}_s \Big) \Big) - 2H^{\text{rest}}_p \sum_{s_1\in\mathcal{S}} (z_{p,d,s_1,a_1}) \quad \forall p\in\mathcal{P},\ d\in[d^{-1},...,d^{e-1}],\ a_1=\{AKC\} \tag{C7.45}$$

$$t^{\text{shi}}_{p,d} - H^{\text{fDS}}_p < 24\,D^{\text{mo}}\,fDH^{\text{sho}}_{p,d}, \qquad \forall p\in\mathcal{P},\ d\in[d^s,...,d^{e-1}] \tag{C7.46}$$

$$H^{\text{fDS}}_p - t^{\text{shi}}_{p,d} \leq 24\,D^{\text{mo}}(1 - fDH^{\text{sho}}_{p,d}), \qquad \forall p\in\mathcal{P},\ d\in[d^s,...,d^{e-1}] \tag{C7.47}$$

$$fD^{\text{sho}}_{p,d} \leq 1 - \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} S^{\text{work}}_s\, z_{p,d,s,a} \qquad \forall p\in\mathcal{P},\ d\in[d^s,...,d^e] \tag{C7.48}$$

$$fDH^{\text{sho}}_{p,d-1} + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} (1 - S^{\text{work}}_s)\, z_{p,d,s,a} \leq\ 1 + fD^{\text{sho}}_{p,d} \quad \forall p\in\mathcal{P},\ d\in[d^{s+1},...,d^e] \tag{C7.49}$$

$$fD^{\text{sho}}_{p,d} = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} (1 - S^{\text{work}}_s)\, z_{p,d,s,a} \qquad \forall p\in\mathcal{P},\ d=\{1\} \tag{C7.50}$$

$$fD^{\text{sho}}_{p,d-1} + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} (1 - S^{\text{work}}_s)\, z_{p,d,s,a} \leq\ 1 + fD^{\text{sho}}_{p,d} \quad \forall p\in\mathcal{P},\ d\in[d^{s+1},...,d^e] \tag{C7.51}$$

$$fDH^{\text{sho}}_{p,d-1} + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \Big(S^{\text{work}}_s\, z_{p,d,s,a} \Big) + fD^{\text{sho}}_{p,d-1} \geq\ fD^{\text{sho}}_{p,d} \quad \forall p\in\mathcal{P},\ d\in[d^{s+1},...,d^e] \tag{C7.52}$$

$$\sum_{d_1=d}^{d+R^{\text{wRow}}_p} fD^{\text{sho}}_{p,d_1} \geq 1 \qquad \forall p\in\mathcal{P},\ d\in[d^{-R^{\text{wRow}}_p},...,d^{e-R^{\text{wRow}}_p}] \tag{C7.53}$$

$$\sum_{d_2=d}^{d+R^{\text{wRow}}_p} fD^{\text{sho}}_{p,d_2} - \sum_{a\in\mathcal{A}} z_{p,d_1,s_1,a} + (1 - fD^{\text{sho}}_{p,d_1}) \geq 1 \quad \forall p\in\mathcal{P},\ d\in[d^{-R^{\text{wRow}}_p},...,d^{e-R^{\text{wRow}}_p}],\ d_1 = d + R^{\text{wRow}}_p,\ s_1=\{SOV\} \tag{C7.54}$$

$$t^{\text{shiF}}_{p,d} \leq \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \Big(H^{\text{endS}}_{s,a}\, z_{p,d,s,a} + H^{\text{startS}}_s\, z_{p,d_1,s,a} \Big) + 24(d_1 - d - 1) + 24\cdot 15 \sum_{s_1\in\mathcal{S}}\sum_{a_1\in\mathcal{A}} S^{\text{fD}}_{s_1}\, z_{p,d_1,s_1,a_1} \quad \forall p\in\mathcal{P},\ d\in[d^{-13},...,d^{e-1}],\ d_1\in[d+1,...,d+14];\ d_1\leq d^{e+3} \tag{C7.55}$$

$$t^{\text{shiF}}_{p,d} \geq \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \Big(H^{\text{endS}}_{s,a}\, z_{p,d,s,a} + H^{\text{startS}}_s\, z_{p,d_1,s,a} \Big) + 24\,(d_1 - d - 1) - 24\cdot 15 \sum_{s_1\in\mathcal{S}}\sum_{a_1\in\mathcal{A}} S^{\text{fD}}_{s_1}\, z_{p,d_1,s_1,a_1} + 24\cdot 15 \left(1 - \sum_{d_2=d+1}^{d_1}\sum_{s_2\in\mathcal{S}}\sum_{a_2\in\mathcal{A}} \Big(1 - S^{\text{fD}}_{s_2} \Big)\ z_{p,d_1,s_2,a_2} - \sum_{s_3\in\mathcal{S}}\sum_{a_3\in\mathcal{A}} S^{\text{fD}}_{s_3}\, z_{p,d_2,s_3,a_3} \right) \quad \forall p\in\mathcal{P},\ d\in[d^{-13},...,d^{e-1}],\ d_1\in[d+1,...,d+14];\ d_1\leq d^{e+3} \tag{C7.56}$$

$$t^{\text{shiF}}_{p,d} - H^{\text{fDS}}_{p} < 24\, D^{\text{mo}}\, fDH_{p,d}, \qquad \forall p \in \mathcal{P},\ d \in [d^{s}, ..., d^{e-1}] \tag{C7.57}$$

$$H^{\text{fDS}}_{p} - t^{\text{shiF}}_{p,d} \leq 24\, D^{\text{mo}}(1 - fDH_{p,d}), \qquad \forall p \in \mathcal{P},\ d \in [d^{s}, ..., d^{e-1}] \tag{C7.58}$$

$$fD_{p,d} \leq 1 - \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} (1 - S^{\text{fD}}_{s})\, z_{p,d,s,a} \qquad \forall p \in \mathcal{P},\ d \in [d^{s}, ..., d^{e}] \tag{C7.59}$$

$$\begin{gathered} fDH_{p,d-1} + \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} S^{\text{fD}}_{s}\, z_{p,d,s,a} \leq 1 + fD_{p,d} \\ \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^{e}] \end{gathered} \tag{C7.60}$$

$$fD_{p,d} = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} S^{\text{fD}}_{s}\, z_{p,d,s,a} \qquad \forall p \in \mathcal{P},\ d = \{1\} \tag{C7.61}$$

$$\begin{gathered} fD_{p,d-1} + \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} S^{\text{fD}}_{s}\, z_{p,d,s,a} \leq 1 + fD_{p,d} \\ \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^{e}] \end{gathered} \tag{C7.62}$$

$$\begin{gathered} fDH_{p,d-1} + \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left(1 - S^{\text{fD}}_{s}\right) z_{p,d,s,a} + fD_{p,d-1} \geq fD_{p,d} \\ \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^{e}] \end{gathered} \tag{C7.63}$$

$$fDH^{\text{start}}_{p,d} = fD_{p,d}, \qquad \forall p \in \mathcal{P},\ d = \{1\} \tag{C7.64}$$

$$fDH^{\text{start}}_{p,d} \geq fD_{p,d} - fD_{p,d-1}, \qquad \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^{e}] \tag{C7.65}$$

$$fDH^{\text{start}}_{p,d} \leq fD_{p,d}, \qquad \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^{e}] \tag{C7.66}$$

$$fDH^{\text{start}}_{p,d} \leq 1 - fD_{p,d-1}, \qquad \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^{e}] \tag{C7.67}$$

$$fDH^{\text{start}}_{p,d} = fD^{\text{start}}_{p,d}, \qquad \forall p \in \mathcal{P}, d \in [d^{1}, ..., d^{e-14}] \tag{C7.68}$$

$$fD^{\text{start}}_{p,d} = 0, \qquad \forall p \in \mathcal{P},\ d \in [d^{s}, ..., d^{-15}] \tag{C7.69}$$

$$fD^{\text{start}}_{p,d} \leq fDH^{\text{start}}_{p,d}, \qquad \forall p \in \mathcal{P},\ d \in [d^{-14}, ..., d^{-1}] \tag{C7.70}$$

$$\begin{gathered} fD^{\text{start}}_{p,d} \geq fDH^{\text{start}}_{p,d} + \sum_{d_1 = d}^{d^{-1}} fD_{p,d_1} - (d^{-1} - d + 1), \\ \forall p \in \mathcal{P},\ d \in [d^{-14}, ..., d^{-1}] \end{gathered} \tag{C7.71}$$

$$fD^{\text{start}}_{p,d} \leq \frac{\sum_{d_1=d}^{d^{-1}} fD_{p,d_1}}{(d^{-1} - d + 1)}, \qquad \forall p \in \mathcal{P}, d \in \{d^{-14}, ..., d^{-1}\} \tag{C7.72}$$

$$fD^{\text{start}}_{p,d} \leq fDH^{\text{start}}_{p,d}, \qquad \forall p \in \mathcal{P},\ d \in \{D^{e-13}, ..., d^{e}\} \tag{C7.73}$$

$$\begin{gathered} \left(1 - fD^{\text{start}}_{p,d}\right) \geq fDH^{\text{start}}_{p,d} + \sum_{d_1 = d}^{d^{e}} fD_{p,d_1} - (d^{e} - d + 1) \\ \forall p \in \mathcal{P},\ d \in [D^{e-13}, ..., d^{e}] \end{gathered} \tag{C7.74}$$

$$fD^{\text{start}}_{p,d} \geq fDH^{\text{start}}_{p,d} - \frac{\sum_{d_1=d}^{d^{e}} fD_{p,d_1}}{(d^{e} - d + 1)}, \qquad \forall p \in \mathcal{P},\ d \in [D^{e-13}, ..., d^{e}] \tag{C7.75}$$

$$45\, fD^{\text{start}}_{p,d} \geq fD^{\text{couS}}_{p,d} + fD^{\text{couR}}_{p,d}, \qquad \forall p \in \mathcal{P},\ d \in [d^{s}, ..., d^{e}] \tag{C7.76}$$

$$\begin{gathered} t^{\text{shiF}}_{p,d-1} \geq H^{\text{fDS}}_{p}\, fD^{\text{couS}}_{p,d} + H^{\text{fDR}}_{p}\, fD^{\text{couR}}_{p,d} - 45\, \left(1 - fD^{\text{start}}_{p,d}\right), \\ \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^{e}] \end{gathered} \tag{C7.77}$$

$$\frac{t^{\text{shiF}}_{p,d-1}}{H^{\text{fDR}}_{p}} - 1 < fD^{\text{couR}}_{p,d} + 45(1 - fD^{\text{start}}_{p,d}) \qquad \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^{e}] \tag{C7.78}$$

$$\frac{t^{\text{shiF}}_{p,d-1}}{H^{\text{fDS}}_{p}} - 1 < fD^{\text{couS}}_{p,d} + \frac{H^{\text{fDR}}_{p}\, fD^{\text{couR}}_{p,d}}{H^{\text{fDS}}_{p}} + 45\left(1 - fD^{\text{start}}_{p,d}\right), \qquad \forall p \in \mathcal{P},\ d \in [d^{s+1}, ..., d^{e}] \tag{C7.79}$$

$$pFD^{\text{u}}_{p} \geq FD^{\text{min}}_{p} - \sum_{d=d^{s}}^{d^{e}} \left(\frac{fD^{\text{couS}}_{p,d}}{2} + fD^{\text{couR}}_{p,d} \right), \qquad \forall p \in \mathcal{P} \tag{C7.80}$$

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} S^{\text{work}}_{s} \left(z_{p,d,s,a} + z_{p,d+1,s,a} \right) \leq 2\, wkd^{\text{work}}_{p,q} \qquad \forall p \in \mathcal{P},\ d \in [d^{-8}, ..., d^{e}],\ q \in \mathcal{Q};\ D^{\text{numL}}_{d} = 6 \tag{C7.81}$$

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} S^{\text{work}}_{s} \left(z_{p,d,s,a} + z_{p,d+1,s,a} \right) \geq\ wkd^{\text{work}}_{p,q} \qquad \forall p \in \mathcal{P},\ d \in [d^{-8}, ..., d^{e}],\ q \in \mathcal{Q};\ D^{\text{numL}}_{d} = 6 \tag{C7.82}$$

$$wkd^{\text{work}}_{p,q} + wkd^{\text{work}}_{p,q+1} \leq pWkd^{\text{work}}_{p,q+1} + 1 + W^{\text{wkd}}_{p,q+1} \qquad \forall p \in \mathcal{P},\ q = \{first\} \tag{C7.83}$$

$$wkd^{\text{work}}_{p,q} + wkd^{\text{work}}_{p,q+1} \leq pWkd^{\text{work}}_{p,q+1} + 1 + \frac{W^{\text{wkd}}_{p,q} + W^{\text{wkd}}_{p,q+1}}{2} \qquad \forall p \in \mathcal{P},\ q \in \mathcal{Q} \backslash \{\text{first}\} \wedge \{\text{last}\} \tag{C7.84}$$

$$1 + p^{\text{row}}_{p,d} \geq \sum_{s \in \mathcal{S}} \sum_{s \in \mathcal{A}} S^{\text{row}}_{s} (z_{p,d-1,s,a} + z_{p,d,s,a}), \qquad \forall p \in \mathcal{P},\ d \in [d^{1}, ...d^{e}] \tag{C7.85}$$

$$1 + p^{\text{row}}_{p,d} \geq \sum_{s \in \mathcal{S}} \sum_{s \in \mathcal{A}} A^{\text{row}}_{a} (z_{p,d-1,s,a} + z_{p,d,s,a}), \qquad \forall p \in \mathcal{P},\ d \in [d^{1}, ...d^{e}] \tag{C7.86}$$

$$1 + p^{\text{row}}_{p,d} \geq \sum_{s \in \mathcal{S}} \sum_{s \in \mathcal{A}} S^{\text{row}}_{s}\ z_{p,d-1,s,a} + A^{\text{row}}_{a}\, z_{p,d,s,a}, \qquad \forall p \in \mathcal{P},\ d \in [d^{1}, ...d^{e}] \tag{C7.87}$$

$$1 + p^{\text{row}}_{p,d} \geq \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} A^{\text{row}}_{a}\ z_{p,d-1,s,a} + S^{\text{row}}_{s}\, z_{p,d,s,a}, \qquad \forall p \in \mathcal{P},\ d \in [d^{1}, ...d^{e}] \tag{C7.88}$$

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} z_{p,d,s,a} \left(K^{\text{shi}}_{s,k} + K^{\text{add}}_{a,k} \right) \geq \sum_{a_1 \in \mathcal{A}} z_{p,d+1,s_1,a_1}, \qquad \forall p \in \mathcal{P},\ d \in [d^{-1}, ..., d^{e-1}],\ k = \{Night\},\ s_1 = \{SOV\} \tag{C7.89}$$

$$\sum_{a \in \mathcal{A}} x_{p,d,s,a} = 0, \qquad \forall p \in \mathcal{P},\ d \in \mathcal{D},\ s \in \mathcal{S};\ S^{\text{ifWi}}_{s} = 1\ \wedge\ WP^{\text{shi}}_{p,d} = \emptyset \tag{C7.90}$$

$$\sum_{a \in \mathcal{A}} x_{p,d,s,a}\ \leq \sum_{a \in \mathcal{A}} WS^{\text{add}}_{w,a}\, WS^{\text{shi}}_{w,s} \qquad \forall p \in \mathcal{P},\ d \in \mathcal{D},\ s \in \mathcal{S},\ w = WP^{\text{shi}}_{p,d};\ S^{\text{ifWi}}_{s} = 1\ \wedge\ WP^{\text{shi}}_{p,d} \neq \emptyset \tag{C7.91}$$

$$\sum_{d_1 = d+1}^{d+5;\, d_1 \leq d^{e}} \sum_{a \in \mathcal{A}} z_{p,d_1,s_1,a} + p^{\text{YL}}_{p,d} \geq \sum_{s \in \mathcal{S}} \sum_{a_1 \in \mathcal{A}} S^{\text{work}}_{s}\, z_{p,d,s,a_1}, \qquad \forall p \in \mathcal{P},\ d \in [d^{-3}, ..., d^{e-2}],\ s_1 = \{\text{FRI\_h}\},\ k = \{YL\};\ D^{\text{numL}}_{d} = 7\ \wedge\ K^{\text{pers}}_{p,k} = 1 \tag{C7.92}$$

$$p^{\mathrm{YL}}_{p,d} = 0 \qquad \forall p \in \mathcal{P},\ d \in [d^s, ..., d^e];\ D^{\mathrm{numL}}_d \neq 7 \vee d < d^{-3} \vee d > d^{e-2} \tag{C7.93}$$

$$\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,d,s,a}\ (RA^{\mathrm{pres}}_a + RS^{\mathrm{pres}}_s)\ K_{p,k} - p^{\mathrm{OLG}} \leq R^{\mathrm{olg}} \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,d,s,a}\ (RA^{\mathrm{pres}}_a + RS^{\mathrm{pres}}_s), \qquad k = \{OLG\} \tag{C7.94}$$

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,d,s,a}\, WS^{\mathrm{add}}_{w,a}\, WS^{\mathrm{shi}}_{w,s} = f^{\mathrm{wGive}}_{p,d}, \quad \forall p \in \mathcal{P},\ d \in \mathcal{D},\ w = \{WP^{\mathrm{shi}}_{p,d}\};\ WP^{\mathrm{prio}}_{p,d} \neq \{*\} \wedge WP^{\mathrm{shi}}_{p,d} \neq \emptyset \wedge WP^{\mathrm{add}}_{p,d} = \emptyset \tag{C7.95}$$

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,d,s,a}\, WA^{\mathrm{add}}_{w,a}\, WA^{\mathrm{shi}}_{w,s} = f^{\mathrm{wGive}}_{p,d}, \quad \forall p \in \mathcal{P},\ d \in \mathcal{D},\ w = \{WP^{\mathrm{add}}_{p,d}\};\ WP^{\mathrm{prio}}_{p,d} \neq \{*\} \wedge WP^{\mathrm{shi}}_{p,d} = \emptyset \wedge WP^{\mathrm{add}}_{p,d} \neq \emptyset \tag{C7.96}$$

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x_{p,d,s,a}\, WS^{\mathrm{add}}_{w_1,a}\, WS^{\mathrm{shi}}_{w_1,s}\, WA^{\mathrm{add}}_{w_2,a}\, WA^{\mathrm{shi}}_{w_2,s} = f^{\mathrm{wGive}}_{p,d}, \qquad \forall p \in \mathcal{P},\ d \in \mathcal{D}, \quad w_1 = \{WP^{\mathrm{shi}}_{p,d}\},\ w_2 = \{WP^{\mathrm{add}}_{p,d}\};\ WP^{\mathrm{prio}}_{p,d} \neq \{*\} \wedge WP^{\mathrm{shi}}_{p,d} \neq \emptyset \wedge WP^{\mathrm{add}}_{p,d} \neq \emptyset \tag{C7.97}$$

$$f^{\mathrm{wGive}}_{p,d} = 0 \qquad \forall p \in \mathcal{P},\ d \in \mathcal{D};\ W^{\mathrm{nStar}}_{p,d} = 0 \tag{C7.98}$$

$$f^{\overline{\mathrm{w}}} = \frac{\sum_{p\in\mathcal{P};\, W^{\mathrm{nStarT}}_p=1}\left(\frac{\sum_{d\in\mathcal{D}} f^{\mathrm{wGive}}_{p,d}}{\sum_{d\in\mathcal{D}} W^{\mathrm{nStar}}_{p,d}}\right)}{\sum_{p\in\mathcal{P}} W^{\mathrm{nStarT}}_p} \tag{C7.99}$$

$$pF^{\mathrm{w}}_{p,t} \geq f^{\overline{\mathrm{w}}} - \frac{\sum_{d\in\mathcal{D}} f^{\mathrm{wGive}}_{p,d}}{\sum_{d\in\mathcal{D}} W^{\mathrm{nStar}}_{p,d}} - W^{\mathrm{step}}_t \qquad \forall p \in \mathcal{P},\ t \in \mathcal{T}; W^{\mathrm{nStarT}}_p = 1 \tag{C7.100}$$

$$f^{\overline{\mathrm{oH}}} = \frac{\sum_{p\in\mathcal{P};\, NR_p\neq 1} \frac{h^{\mathrm{ov}}_p}{H^{\mathrm{targ}}_p}}{\sum_{p\in\mathcal{P}}(1 - NR_p)} \tag{C7.101}$$

$$pF^{\mathrm{oH}}_{p,t} \geq \frac{h^{\mathrm{ov}}_p}{H^{\mathrm{targ}}_p} - f^{\overline{\mathrm{oH}}} - OH^{\mathrm{step}}_t, \qquad \forall p \in \mathcal{P},\ t \in \mathcal{T};\ NR_p \neq 1 \tag{C7.102}$$

$$f^{\overline{\mathrm{cou}}}_c = \frac{\sum_{p\in\mathcal{P}}\left(\frac{\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C^{\mathrm{pos}}_{p,c,s,a}\, x^{\mathrm{help}}_{p,s,a}}{N^{\mathrm{WorkD}}_p}\right)}{\sum_{p\in\mathcal{P}} C^{\mathrm{emp}}_{p,c}} \qquad \forall c \in \mathcal{C}^{\mathrm{f}} \tag{C7.103}$$

$$pF^{\mathrm{cou}}_{p,c,t} \geq \frac{\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} C^{\mathrm{pos}}_{p,c,s,a}\, x^{\mathrm{help}}_{p,s,a}}{N^{\mathrm{WorkD}}_p} - f^{\overline{\mathrm{cou}}}_c - C^{\mathrm{step}}_t, \qquad \forall p \in \mathcal{P}, c \in \mathcal{C}^{\mathrm{f}}, t \in \mathcal{T} \tag{C7.104}$$

B Results

B.0.1 Output sheets from the model

The output shown is generated based on the September instance.

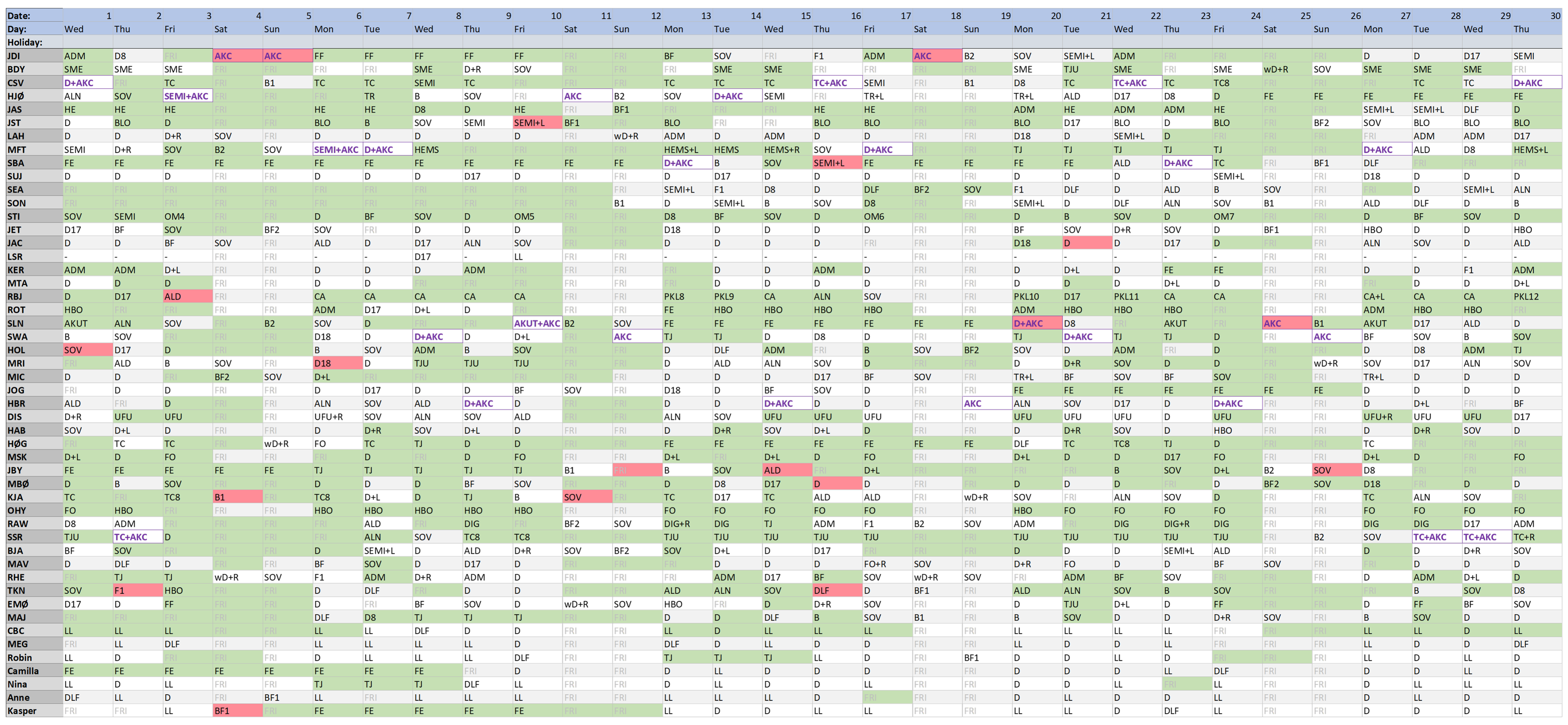

Figure B.1: The *Schedule* sheet. All shifts+add-ons given to each physician on all days. Some shifts and add-ons from the data are modified to a more presentable form and some even removed. For example the shift "FRI_h" and "FRI_w" is changed to "FRI" and the add-on "ingen" is completely removed. Cells in the plan are filled with a green color if a wish is fulfilled and red if not fulfilled. Lastly, the shift/add-on "AKC" is marked with a purple bold font and a purple border.

Figure B.2: The *Wish Status* sheet. Consist of all wishes for each physician each day. Not fulfilled wishes are marked with red. All the way to the right are all the wishes not granted counted.

KPI	Min	Ave	Max	Total
Over Hours [Hours]	0	0.52	6.54	25.6066319
Under Hours [Hours]	0	4.97	38.065	243.565
Contracted Over Hours [Hours]	0	0.05	0.78	2.25456
Contracted Under Hours [Hours]	0	1.01	43.1666	49.2703
Amount of FRI_h given [Count]	0	2.61	9	128
Amount of FRI_w given [Count]	1	6.16	8	302
Amount of FRI given [Count]	4	8.78	14	430
Number of weekends working	0	0.88	3	44

Priorities				
	P8	TC8	D8	Other
t_8	0	6	16	0
	RLG	Other		
t_LL	65	1		
	D	Other		
t_SPLG	272	94		
	FRI_w	Other		
t_FRI	310	146		
	-	Other		
t_LSR	20	8		
	D	Other		
t_ALL	324	1176		
	YL	Other		
t_DLF	18	4		

Total Antal Ønsker	Ønsker Opfyldt	Procent Ønsker Opfyldtr	
76	56	74%	*her tæller stjernemarkerede ønsker ikke med

Min Procent	Ave Procent	Max Procemt	
50%	74%	100%	*her tæller stjernemarkerede ønsker ikke med

Emp	JDI	CSV	JST	SBA	JAC	RBJ	SLN	HOL	MRI	DIS	JBY	MBØ	KJA	SSR	MAV	TKN	MAJ	CBC	Kasper
WishEmp	7	1	2	2	6	3	5	3	4	1	7	2	5	1	1	10	1	13	2
WishEmpFulfilled	4	1	1	1	5	2	3	2	3	1	4	1	3	1	1	8	1	13	1
WishEmpFulfilled Percentage	57%	100%	50%	50%	83%	67%	60%	67%	75%	100%	57%	50%	60%	100%	100%	80%	100%	100%	50%

Total tilstedeværelsesvagter:	94
Filled by OLG:	51
% Filled by OLG:	54.26

Figure B.3: The *KPI* sheet. Highlights minimums, averages and maximums for over and under hours, contracted over/under hours, "FRI_h", "FRI_w", "FRI", and number of weekends. This sheets also provides details about the priorities from *Data sheet 7: Priorities*. It also visualizes the overview of wishes fulfilled both for each physicians and overall. Lastly, the chief physician fraction is also presented in this sheet.

week number	35	35	35	35	35	36	36	36	36	36	36	36	37	37	37	37	37	37	37	38	38	38	38	38	38	38	39	39	39	39
date number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
counter/week day	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu
t_SPLG demand	16	14	13	0	0	16	17	16	17	15	0	0	16	17	16	17	16	0	0	16	17	16	17	15	0	0	16	17	16	17
t_SPLG actual	16	14	13	1	1	16	17	16	17	15	2	2	16	17	16	17	16	2	1	16	17	16	17	15	2	2	16	17	16	17
t_RLG demand	2	2	2	0	0	2	2	2	2	2	0	0	2	2	2	2	2	0	0	2	2	2	2	2	0	0	2	2	2	2
t_RLG actual	3	5	4	1	1	4	3	3	4	5	0	0	5	6	6	7	6	0	1	7	7	7	5	3	0	0	7	7	7	6
t_Børn demand	3	3	3	0	0	3	3	3	3	3	0	0	3	3	3	3	3	0	0	3	3	3	3	3	0	0	3	3	3	3
t_Børn actual	3	3	4	0	0	6	6	3	3	3	0	1	4	7	4	4	6	1	0	7	6	7	4	4	1	0	3	4	4	5
t_U demand	2	2	2	0	0	2	2	2	2	2	0	0	2	2	2	2	2	0	0	2	2	2	2	2	0	0	2	2	2	2
t_U actual	7	7	5	0	1	6	6	7	7	5	1	2	7	7	6	8	7	0	1	8	9	8	9	4	1	1	8	7	9	7
t_S demand	2	2	2	0	0	2	2	2	2	2	0	0	2	2	2	2	2	0	0	2	2	2	2	2	0	0	2	2	2	2
t_S actual	3	5	4	0	0	6	7	5	3	2	0	0	4	5	4	3	5	0	0	4	7	2	2	4	0	1	3	7	3	4
t_F demand	2	2	2	0	0	2	2	2	2	2	0	0	2	2	2	2	2	0	0	2	2	2	2	2	0	0	2	2	2	2
t_F actual	7	4	5	1	0	6	6	5	8	9	1	0	6	7	8	8	6	2	0	6	3	8	7	7	1	0	7	5	6	7
t_D17 demand	2	2	0	0	0	0	2	2	2	0	0	0	0	2	2	2	0	0	0	0	2	2	2	0	0	0	0	2	2	2
t_D17 actual	2	2	0	0	0	0	2	2	2	0	0	0	0	2	2	2	0	0	0	0	2	2	2	0	0	0	0	2	2	2
t_D18 demand	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0
t_D18 actual	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0
t_D20 demand	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_D20 actual	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_D20Børn demand	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1
t_D20Børn actual	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1
t_D22 demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_D22 actual	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_R demand	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_R actual	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1
t_wD demand	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0
t_wD actual	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0
t_AKC demand	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_AKC actual	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_LL demand	3	3	3	0	0	3	3	3	3	3	0	0	3	3	3	3	3	0	0	3	3	3	3	3	0	0	3	3	3	3
t_LL actual	3	3	3	0	0	3	3	3	3	3	0	0	3	3	3	3	3	0	0	3	3	3	3	3	0	0	3	3	3	3
t_SEMI demand	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_SEMI actual	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_8 demand	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_8 actual	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_B demand	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_B actual	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_B1 demand	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0
t_B1 actual	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0
t_B2 demand	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0
t_B2 actual	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0
t_DLF demand	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_DLF actual	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_BF demand	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_BF actual	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_BF1 demand	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0
t_BF1 actual	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0
t_BF2 demand	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0
t_BF2 actual	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0
t_ALD demand	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_ALD actual	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_ALN demand	1	1	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	1	1
t_ALN actual	1	1	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	1	1
t_HBO demand	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_HBO actual	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1
t_FRI demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_FRI actual	9	8	11	41	41	7	7	8	7	10	41	41	6	4	3	4	4	41	41	2	3	1	5	8	41	41	6	4	5	6
t_ALL demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_ALL actual	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
t_LANG demand	4	4	2	0	0	4	4	4	4	2	0	0	4	4	4	4	2	0	0	4	4	4	4	2	0	0	4	4	4	4
t_LANG actual	4	4	2	0	0	4	4	4	4	2	0	0	4	4	4	4	2	0	0	4	4	4	4	2	0	0	4	4	4	4
t_NAT demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_NAT actual	3	3	2	2	2	3	3	3	3	2	2	2	3	3	3	3	2	2	2	3	3	3	3	2	2	2	3	3	3	3
t_DLFYL demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_DLFYL actual	1	1	1	0	0	1	1	1	1	1	0	0	1	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	1
t_F1 demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_F1 actual	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0
t_ADM demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_ADM actual	2	2	0	0	0	1	1	1	2	0	0	0	1	1	2	2	1	0	0	3	1	3	1	0	0	0	1	2	2	2
t_TR demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_TR actual	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0	0	1	0	0	0
t_DIG demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_DIG actual	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	1	1	1	0	0	1	1	0	0
t_SME demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_SME actual	1	1	1	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	1	0	0	1	1	1	0
t_TC demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_TC actual	1	2	3	0	0	2	2	0	2	1	0	0	2	1	2	1	0	0	0	0	2	2	1	2	0	0	2	2	2	1
t_FO demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_FO actual	1	0	1	0	0	1	0	0	0	1	0	0	1	1	1	1	3	0	0	0	2	1	1	2	0	0	1	1	1	2
t_BLO demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_BLO actual	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	1	0	1	0	0	0	1	1	1
t_OMS demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_OMS actual	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
t_CA demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_CA actual	0	0	0	0	0	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	1	1	0
t_AKUT demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_AKUT actual	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
t_HEMS demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_HEMS actual	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
t_UFU demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_UFU actual	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1	0	1	0	0	1	1	1	0
t_CSV demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_CSV actual	4	4	3	4	4	3	4	4	3	3	4	4	4	4	4	4	4	4	4	4	4	3	4	3	4	4	4	4	4	4
t_LSR demand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_LSR actual	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure B.4: The *Demand* sheet. Specifies the planned demand vs. the actual amount assigned in the schedule for all counters on all days.

Personale/Tællere	Total hours	number of weekends working	nDays t_F1	nDays t_ADM	nDays t_TR	% t_DIG	% t_SME	% t_TC	% t_FO	% t_BLO	% t_OMS	% t_CA	% t_AKUT	% t_HEMS	% t_UFU	% t_CSV	% t_LSR	% Resterende
JDI contracted	146.52	-	1	3	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
JDI actual	146.75	2	1	3	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
BDY contracted	113.96	-	0	0	0	0%	85%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	15%
BDY actual	120.5	1	0	0	0	0%	85%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	22%
CSV contracted	170.94	-	0	0	0	0%	0%	61%	0%	0%	0%	0%	0%	0%	0%	39%	0%	0%
CSV actual	158.5	2	0	0	0	0%	0%	61%	0%	0%	0%	0%	0%	0%	0%	14%	0%	18%
HJØ contracted	170.94	-	0	0	2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
HJØ actual	165.75	1	0	0	3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	90%
JAS contracted	170.94	-	0	3	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
JAS actual	165.5	1	0	3	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	96%
JST contracted	170.94	-	0	0	0	0%	0%	0%	0%	50%	0%	0%	0%	0%	0%	0%	0%	50%
JST actual	173.5	2	0	0	0	0%	0%	0%	0%	48%	0%	0%	0%	0%	0%	0%	0%	53%
LAH contracted	170.94	-	0	4	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
LAH actual	171.75	1	0	4	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	101%
MFT contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
MFT actual	170	1	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	99%
SBA contracted	170.94	-	0	0	0	0%	0%	9%	0%	0%	0%	0%	0%	0%	0%	0%	0%	91%
SBA actual	168.375	1	0	0	0	0%	0%	9%	0%	0%	0%	0%	0%	0%	0%	0%	0%	84%
SUJ contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
SUJ actual	171.5	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
SEA contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
SEA actual	152.5	1	2	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	89%
SON contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
SON actual	154.5	2	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	90%
STI contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
STI actual	167.596632	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	97%
JET contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
JET actual	172	2	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	101%
JAC contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
JAC actual	171.75	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
LSR contracted	0	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%
LSR actual	15.25	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	#VALUE!
KER contracted	170.94	-	1	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
KER actual	160.25	0	1	5	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	93%
MTA contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
MTA actual	137	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	80%
RBJ contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
RBJ actual	171.75	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	101%
ROT contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
ROT actual	132.875	0	0	3	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	76%
SLN contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
SLN actual	171.125	3	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	99%
SWA contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
SWA actual	169.5	2	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	99%
HOL contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
HOL actual	170.25	1	0	4	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
MRI contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
MRI actual	172	1	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	101%
MIC contracted	170.94	-	0	0	1	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
MIC actual	171.5	1	0	0	2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	96%
JOG contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
JOG actual	172.375	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
HBR contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
HBR actual	167.5	1	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	98%
DIS contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	57%	0%	0%	43%
DIS actual	171.25	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	57%	0%	0%	43%
HAB contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
HAB actual	157.75	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	92%
HØG contracted	170.94	-	0	0	0	0%	0%	39%	0%	0%	0%	0%	0%	0%	0%	0%	0%	61%
HØG actual	153.875	1	0	0	0	0%	0%	39%	0%	0%	0%	0%	0%	0%	0%	0%	0%	46%
MSK contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
MSK actual	159	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	93%
JBY contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
JBY actual	160.125	2	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	89%
MBØ contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
MBØ actual	171.5	1	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
KJA contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
KJA actual	172	2	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	101%
OHY contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
OHY actual	157.5	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	92%
RAW contracted	170.94	-	0	4	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
RAW actual	169.25	2	1	4	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	99%
SSR contracted	170.94	-	0	0	0	0%	0%	50%	0%	0%	0%	0%	0%	0%	0%	0%	0%	50%
SSR actual	170	1	0	0	0	0%	0%	50%	0%	0%	0%	0%	0%	0%	0%	0%	0%	48%
BJA contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
BJA actual	170.5	1	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
MAV contracted	170.94	-	0	0	0	0%	0%	0%	10%	0%	0%	0%	0%	0%	0%	0%	0%	90%
MAV actual	167.25	0	0	0	0	0%	0%	0%	9%	0%	0%	0%	0%	0%	0%	0%	0%	89%
RHE contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
RHE actual	171	2	1	5	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
TKN contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
TKN actual	168.5	1	1	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	99%
EMØ contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
EMØ actual	160.125	1	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	92%
MAJ contracted	170.94	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
MAJ actual	167.5	1	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	98%
CBC contracted	162.8	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
CBC actual	163	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
MEG contracted	162.8	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
MEG actual	166.5	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	102%
Robin contracted	162.8	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
Robin actual	163.5	1	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
Camilla contracted	162.8	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
Camilla actual	163	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
Nina contracted	162.8	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
Nina actual	163	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
Anne contracted	162.8	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
Anne actual	163.5	1	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
Kasper contracted	162.8	-	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
Kasper actual	163.5	1	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	101%

Figure B.5: The *Hours and Contract* sheet. Specifies the contracted and actual assigned total hours, and their respective contracts both including number of days and percentage contracts. It also specifies the number of weekends working.

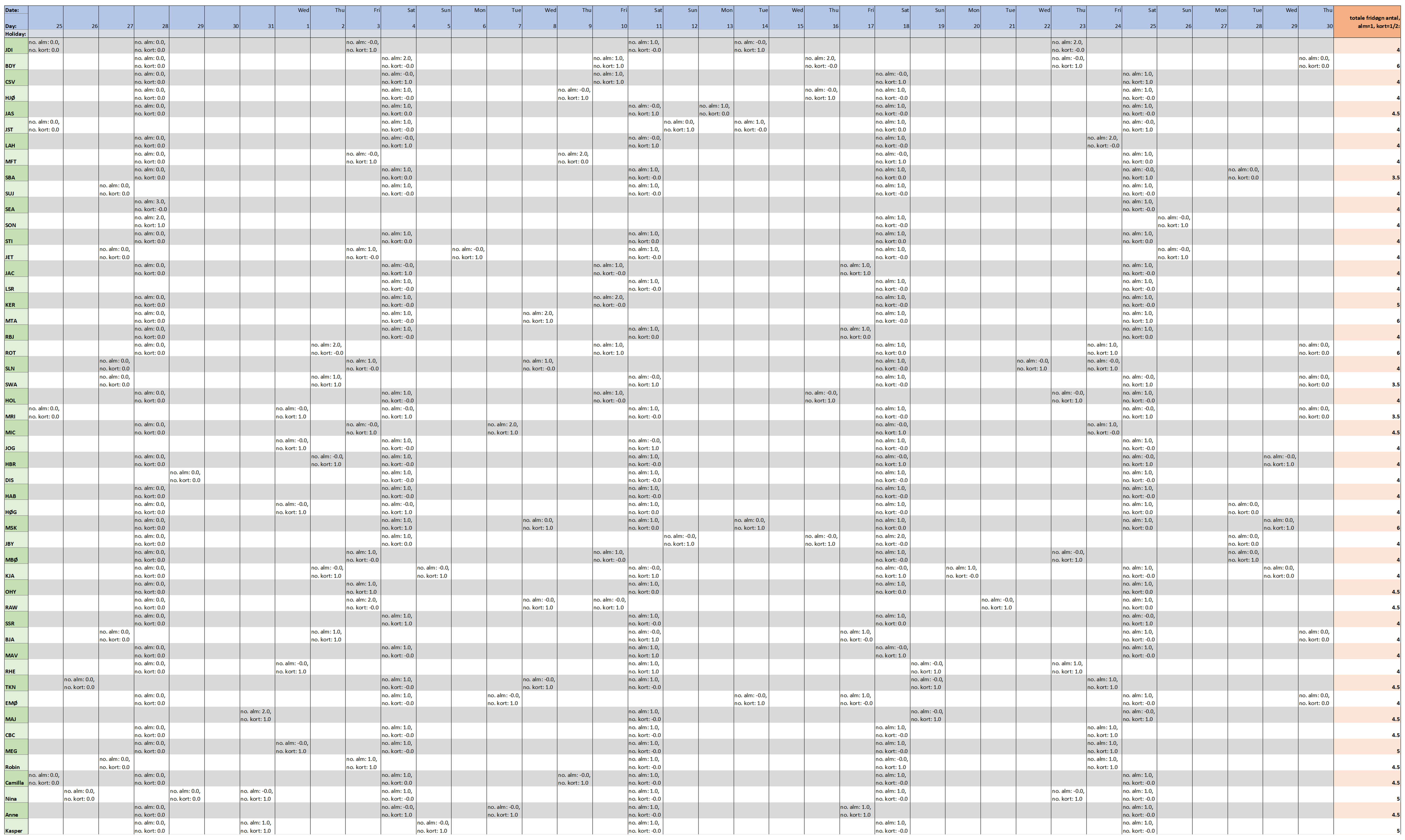

Figure B.6: The *Rest Period* sheet shows how many rest periods are given to each physician each day and the total number of rest periods assigned for the entire month.

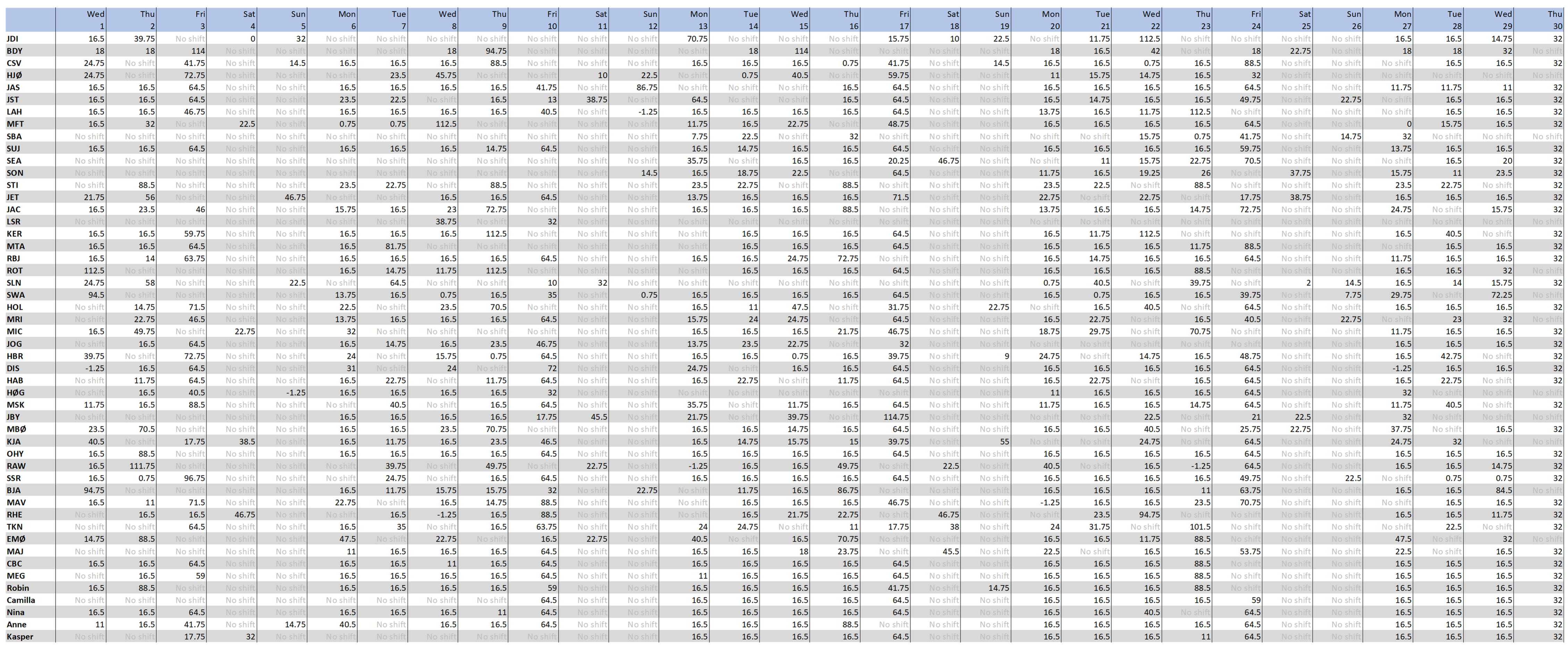

Figure B.7: The *Time to next working shifts* sheets. Provides the actual amount of hours to next working shift for each employee on each day. "No shift" highlights an 'off day' shift.

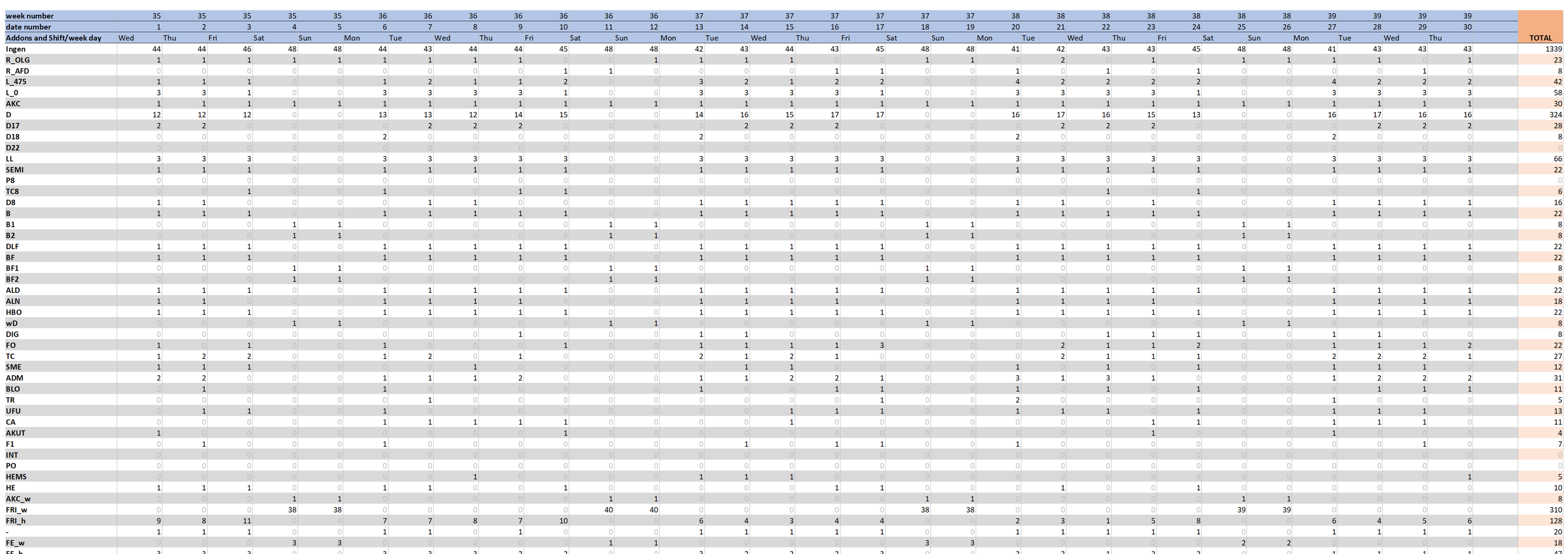

Figure B.8: The *Number of shifts pr day* sheet. This sheet contains the number of each shift given per day and the total number assigned the entire month.

Emp/Shi	Ingen	R_OLG	R_AFD	L_475	L_0	AKC	D	D17	D18	D22	LL	SEMI	P8	TC8	D8	B	B1	B2	DLF	BF	BF1	BF2	ALD	ALN	HBO	wD	DIG	FO	TC	SME	ADM	BLO	TR	UFU	CA	AKUT	F1	INT	PO	HEMS	HE	AKC_w	FRI_w	RI_	-	FE_w	E_I	FF	ASP	KO	PKL	TJ	TJU	OM	SOV	SYG	DS	DB	DBF	LLBF	LLB
JDI	25	0	0	1	1	3	2	1	0	0	0	2	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	1	0	0	0	0	3	4	4	0	0	0	5	0	0	0	0	0	0	2	0	0	0	0	0	0
BDY	28	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0
CSV	26	0	0	0	0	4	2	0	0	0	0	2	0	1	1	0	2	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	6	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HJØ	24	0	0	2	1	3	2	1	0	0	0	2	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	1	4	3	0	2	4	0	0	0	0	0	0	0	3	0	0	0	0	0	0
JAS	27	0	0	2	1	0	2	0	0	0	0	2	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	10	0	7	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
JST	28	0	0	1	1	0	3	1	0	0	0	2	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	6	2	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
LAH	25	2	0	1	2	0	13	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	6	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
MFT	22	2	0	2	0	4	4	0	0	0	0	2	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	6	2	0	0	0	0	0	0	0	5	0	0	3	0	0	0	0	0	0
SBA	26	0	0	1	1	2	2	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0	6	11	0	0	0	0	0	0	0	1	0	0	0	0	0	0
SUJ	26	0	0	1	3	0	18	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SEA	26	0	0	2	2	0	3	0	0	0	0	2	0	0	1	1	0	0	2	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	5	9	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
SON	26	0	0	2	2	0	3	0	0	0	0	2	0	0	1	2	2	0	2	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	8	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
STI	30	0	0	0	0	0	7	0	0	0	0	1	0	0	1	1	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	4	5	0	0	0	0	0	0
JET	27	0	1	0	2	0	11	1	1	0	0	0	0	0	0	0	0	0	0	2	1	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	1	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
JAC	27	0	0	0	3	0	11	2	1	0	0	0	0	0	0	0	0	0	0	1	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	1	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
LSR	29	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KER	28	0	0	2	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	1	0	0	0	0	0	8	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MTA	28	0	0	2	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBJ	27	0	0	1	2	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	5	0	0	0	1	0	0	0	0	0	0
ROT	28	0	0	1	1	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	8	5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SLN	26	0	0	0	1	3	3	1	0	0	0	0	0	0	1	0	1	2	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	1	1	4	0	2	5	0	0	0	0	0	0	0	3	0	0	0	0	0	0
SWA	24	0	0	1	1	4	8	0	1	0	0	0	0	0	1	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	6	1	0	0	0	0	0	0	0	5	0	0	3	0	0	0	0	0	0
HOL	28	0	0	0	2	0	5	1	0	0	0	0	0	0	1	3	0	0	1	0	0	1	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	6	2	0	0	0	0	0	0	0	1	0	0	5	0	0	0	0	0	0
MRI	26	2	0	0	2	0	7	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	2	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	1	0	0	0	0	0	0	0	0	3	0	5	0	0	0	0	0	0
MIC	26	0	0	3	1	0	9	1	0	0	0	0	0	0	0	0	0	0	0	3	0	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	5	5	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
JOG	28	0	0	0	2	0	11	1	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	1	0	2	5	0	0	0	0	0	0	0	2	0	0	0	0	0	0
HBR	24	0	0	1	1	4	12	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	7	2	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
DIS	26	3	0	0	1	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
HAB	23	4	0	3	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
HØG	28	1	0	0	1	0	3	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	5	4	0	2	5	0	0	0	0	2	0	0	0	0	0	0	0	0	0
MSK	24	0	0	5	1	0	12	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
JBY	28	0	0	2	0	0	2	0	0	0	0	0	0	0	1	2	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	6	0	2	3	0	0	0	0	5	0	0	3	0	0	0	0	0	0
MBØ	28	0	0	0	2	0	13	1	1	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	2	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
KJA	27	1	0	1	1	0	3	1	0	0	0	0	0	2	0	1	1	0	0	0	0	0	2	2	0	1	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	5	3	0	0	0	0	0	0	0	1	0	0	4	0	0	0	0	0	0
OHY	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RAW	27	2	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	8	0	0	0	4	0	0	0	0	0	1	0	0	0	0	0	4	5	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0	0
SSR	26	1	0	0	0	3	1	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	1	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	7	1	0	0	0	0	0	0	0	0	11	0	2	0	0	0	0	0	0
BJA	24	0	2	3	1	0	11	1	0	0	0	2	0	0	0	0	0	0	0	1	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	2	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
MAV	26	0	2	0	2	0	13	1	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	2	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
RHE	25	3	0	1	1	0	5	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	5	0	0	0	0	0	1	0	0	0	0	0	4	4	0	0	0	0	0	0	0	2	0	0	4	0	0	0	0	0	0
TKN	28	0	0	0	2	0	4	0	0	0	0	0	0	0	1	2	0	0	2	0	1	0	2	2	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	7	2	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
EMØ	26	0	2	1	1	0	9	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	2	0	0	0	3	0	0	0	0	1	0	4	0	0	0	0	0	0
MAJ	27	0	1	0	2	0	7	0	0	0	0	0	0	0	1	3	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	3	0	0	0	0	0	0	0	3	0	0	4	0	0	0	0	0	0
CBC	29	0	0	0	1	0	4	0	0	0	16	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MEG	27	0	0	0	3	0	8	0	0	0	9	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Robin	29	0	0	0	1	0	8	0	0	0	8	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	2	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
Camilla	29	0	0	0	1	0	11	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	1	0	2	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Nina	29	0	0	0	1	0	7	0	0	0	10	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	1	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
Anne	29	0	0	0	1	0	8	0	0	0	11	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kasper	29	0	0	0	1	0	6	0	0	0	8	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	2	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure B.9: the *Number of shifts pr physician* sheet. This sheet contains the number of each shifts given to each physician.

	0 KO	PKL	OM
JDI	0	0	0
BDY	0	0	1
CSV	0	0	0
HJØ	0	0	0
JAS	0	0	0
JST	0	0	0
LAH	0	0	0
MFT	0	0	0
SBA	0	0	0
SUJ	5	0	0
SEA	0	0	0
SON	0	0	0
STI	0	0	7
JET	0	0	0
JAC	0	0	0
LSR	0	0	0
KER	0	0	0
MTA	0	0	0
RBJ	0	12	0
ROT	0	0	0
SLN	8	0	0
SWA	0	0	0
HOL	0	0	0
MRI	0	0	0
MIC	0	0	0
JOG	0	0	0
HBR	0	0	0
DIS	0	0	0
HAB	0	0	0
HØG	0	0	0
MSK	0	0	0
JBY	0	0	0
MBØ	0	0	0
KJA	0	0	0
OHY	0	0	0
RAW	0	0	0
SSR	0	0	0
BJA	0	0	0
MAV	0	0	0
RHE	0	0	0
TKN	0	0	0
EMØ	0	0	2
MAJ	0	0	0
CBC	0	0	0
MEG	0	0	0
Robin	0	0	0
Camilla	0	0	0
Nina	0	0	0
Anne	0	0	1

Figure B.10: The *Historical shifts* sheet. This sheet includes the number of a specific shift, a physician has been assigned in total, in the current year.

Emp		t_D17	t_D18	t_R	t_AKC	t_SEMI	t_8	t_ALD	t_ALN	t_LANG	t_NAT	t_DLFYL	t_R+t_NAT	t_R+t_NAT som %
JDI	Antal Arbejdsdage	21	21	21	21	21	21	21	21	21	21	21	21	
	Antal Vagter	1	0	0	3	2	1	0	0	1	2	0	2	10%
BDY	Antal Arbejdsdage	24	24	24	24	24	24	24	24	24	24	24	24	
	Antal Vagter	0	0	2	0	0	0	0	0	0	0	0	2	8%
CSV	Antal Arbejdsdage	22	22	22	22	22	22	22	22	22	22	22	22	
	Antal Vagter	0	0	0	4	2	2	0	0	0	0	0	0	0%
HJØ	Antal Arbejdsdage	22	22	22	22	22	22	22	22	22	22	22	22	
	Antal Vagter	1	0	0	3	2	1	1	1	1	3	0	3	14%
JAS	Antal Arbejdsdage	21	21	21	21	21	21	21	21	21	21	21	21	
	Antal Vagter	0	0	0	0	2	1	0	0	1	0	0	0	0%
JST	Antal Arbejdsdage	24	24	24	24	24	24	24	24	24	24	24	24	
	Antal Vagter	1	0	0	0	2	0	0	0	1	2	0	2	8%
LAH	Antal Arbejdsdage	27	27	27	27	27	27	27	27	27	27	27	27	
	Antal Vagter	1	1	2	0	1	0	0	0	2	0	0	2	7%
MFT	Antal Arbejdsdage	22	22	22	22	22	22	22	22	22	22	22	22	
	Antal Vagter	0	0	2	4	2	1	1	0	0	1	0	3	14%
SBA	Antal Arbejdsdage	10	10	10	10	10	10	10	10	10	10	10	10	
	Antal Vagter	0	0	0	2	1	0	1	0	1	1	0	1	10%
SUJ	Antal Arbejdsdage	30	30	30	30	30	30	30	30	30	30	30	30	
	Antal Vagter	2	1	0	0	1	0	0	0	3	0	0	0	0%
SEA	Antal Arbejdsdage	18	18	18	18	18	18	18	18	18	18	18	18	
	Antal Vagter	0	0	0	0	2	1	1	1	2	3	2	3	17%
SON	Antal Arbejdsdage	18	18	18	18	18	18	18	18	18	18	18	18	
	Antal Vagter	0	0	0	0	2	1	1	1	2	3	2	3	17%
STI	Antal Arbejdsdage	18	18	18	18	18	18	18	18	18	18	18	18	
	Antal Vagter	0	0	0	0	1	1	0	0	0	4	0	4	22%
JET	Antal Arbejdsdage	27	27	27	27	27	27	27	27	27	27	27	27	
	Antal Vagter	1	1	1	0	0	0	0	0	2	3	0	4	15%
JAC	Antal Arbejdsdage	30	30	30	30	30	30	30	30	30	30	30	30	
	Antal Vagter	2	1	0	0	0	0	2	2	3	3	0	3	10%
LSR	Antal Arbejdsdage	30	30	30	30	30	30	30	30	30	30	30	30	
	Antal Vagter	1	0	0	0	0	0	0	0	1	0	0	0	0%
KER	Antal Arbejdsdage	26	26	26	26	26	26	26	26	26	26	26	26	
	Antal Vagter	0	0	0	0	0	0	0	0	2	0	0	0	0%
MTA	Antal Arbejdsdage	25	25	25	25	25	25	25	25	25	25	25	25	
	Antal Vagter	0	0	0	0	0	0	0	0	2	0	0	0	0%
RBJ	Antal Arbejdsdage	30	30	30	30	30	30	30	30	30	30	30	30	
	Antal Vagter	2	0	0	0	0	0	1	1	2	1	0	1	3%
ROT	Antal Arbejdsdage	24	24	24	24	24	24	24	24	24	24	24	24	
	Antal Vagter	1	0	0	0	0	0	0	0	2	0	0	0	0%
SLN	Antal Arbejdsdage	19	19	19	19	19	19	19	19	19	19	19	19	
	Antal Vagter	1	0	0	3	0	1	1	1	1	3	0	3	16%
SWA	Antal Arbejdsdage	26	26	26	26	26	26	26	26	26	26	26	26	
	Antal Vagter	0	1	0	4	0	1	0	0	2	3	0	3	12%
HOL	Antal Arbejdsdage	24	24	24	24	24	24	24	24	24	24	24	24	
	Antal Vagter	1	0	0	0	0	1	0	0	2	4	0	4	17%
MRI	Antal Arbejdsdage	27	27	27	27	27	27	27	27	27	27	27	27	
	Antal Vagter	1	1	2	0	0	0	2	2	2	3	0	5	19%
MIC	Antal Arbejdsdage	23	23	23	23	23	23	23	23	23	23	23	23	
	Antal Vagter	1	0	0	0	0	0	0	0	2	4	0	4	17%
JOG	Antal Arbejdsdage	23	23	23	23	23	23	23	23	23	23	23	23	
	Antal Vagter	1	1	0	0	0	0	0	0	2	2	0	2	9%
HBR	Antal Arbejdsdage	27	27	27	27	27	27	27	27	27	27	27	27	
	Antal Vagter	1	0	0	4	0	0	2	2	2	3	0	3	11%
DIS	Antal Arbejdsdage	27	27	27	27	27	27	27	27	27	27	27	27	
	Antal Vagter	1	0	3	0	0	0	1	2	1	2	0	5	19%
HAB	Antal Arbejdsdage	30	30	30	30	30	30	30	30	30	30	30	30	
	Antal Vagter	0	0	4	0	0	0	0	0	3	0	0	4	13%
HØG	Antal Arbejdsdage	15	15	15	15	15	15	15	15	15	15	15	15	
	Antal Vagter	0	0	1	0	0	1	0	0	1	0	0	1	7%
MSK	Antal Arbejdsdage	26	26	26	26	26	26	26	26	26	26	26	26	
	Antal Vagter	1	0	0	0	0	0	0	0	6	0	0	0	0%
JBY	Antal Arbejdsdage	18	18	18	18	18	18	18	18	18	18	18	18	
	Antal Vagter	0	0	0	0	0	1	1	0	2	3	0	3	17%
MBØ	Antal Arbejdsdage	23	23	23	23	23	23	23	23	23	23	23	23	
	Antal Vagter	1	1	0	0	0	1	0	0	2	3	0	3	13%
KJA	Antal Arbejdsdage	29	29	29	29	29	29	29	29	29	29	29	29	
	Antal Vagter	1	0	1	0	0	2	2	2	2	3	0	4	14%
OHY	Antal Arbejdsdage	29	29	29	29	29	29	29	29	29	29	29	29	
	Antal Vagter	0	0	0	0	0	0	0	0	0	0	0	0	0%
RAW	Antal Arbejdsdage	23	23	23	23	23	23	23	23	23	23	23	23	
	Antal Vagter	1	0	2	0	0	1	1	0	1	2	0	4	17%
SSR	Antal Arbejdsdage	24	24	24	24	24	24	24	24	24	24	24	24	
	Antal Vagter	0	0	1	3	0	2	0	1	0	2	0	3	13%
BJA	Antal Arbejdsdage	24	24	24	24	24	24	24	24	24	24	24	24	
	Antal Vagter	1	0	2	0	2	0	2	0	2	2	0	4	17%
MAV	Antal Arbejdsdage	25	25	25	25	25	25	25	25	25	25	25	25	
	Antal Vagter	1	0	2	0	0	0	0	0	2	2	1	4	16%
RHE	Antal Arbejdsdage	28	28	28	28	28	28	28	28	28	28	28	28	
	Antal Vagter	1	0	3	0	0	0	0	0	2	2	0	5	18%
TKN	Antal Arbejdsdage	23	23	23	23	23	23	23	23	23	23	23	23	
	Antal Vagter	0	0	0	0	0	1	2	2	2	4	2	4	17%
EMØ	Antal Arbejdsdage	23	23	23	23	23	23	23	23	23	23	23	23	
	Antal Vagter	1	0	2	0	0	0	0	0	2	2	0	4	17%
MAJ	Antal Arbejdsdage	23	23	23	23	23	23	23	23	23	23	23	23	
	Antal Vagter	0	0	1	0	0	1	0	0	2	3	2	4	17%
CBC	Antal Arbejdsdage	26	26	26	26	26	26	26	26	26	26	26	26	
	Antal Vagter	0	0	0	0	0	0	0	0	1	0	1	0	0%
MEG	Antal Arbejdsdage	30	30	30	30	30	30	30	30	30	30	30	30	
	Antal Vagter	0	0	0	0	0	0	0	0	3	0	3	0	0%
Robin	Antal Arbejdsdage	26	26	26	26	26	26	26	26	26	26	26	26	
	Antal Vagter	0	0	0	0	0	0	0	0	1	0	1	0	0%
Camilla	Antal Arbejdsdage	22	22	22	22	22	22	22	22	22	22	22	22	
	Antal Vagter	0	0	0	0	0	0	0	0	1	0	1	0	0%
Nina	Antal Arbejdsdage	29	29	29	29	29	29	29	29	29	29	29	29	
	Antal Vagter	0	0	0	0	0	0	0	0	1	0	1	0	0%
Anne	Antal Arbejdsdage	29	29	29	29	29	29	29	29	29	29	29	29	
	Antal Vagter	0	0	0	0	0	0	0	0	1	0	1	0	0%
Kasper	Antal Arbejdsdage	23	23	23	23	23	23	23	23	23	23	23	23	
	Antal Vagter	0	0	0	0	0	0	0	0	1	0	1	0	0%

Figure B.11: The *Fairness counter* sheet. Contains the fairness counters and the number of shifts a physician has received for each counter, together with the total number of available working days. A consolidation of the "t_R" and "t_NAT" is also shown (this additional fairness counter was not presented in the model).

Technical
University of
Denmark

Akademivej, Building 358
2800 Kgs. Lyngby
Tlf. 4525 1700

www.man.dtu.dk