MASTERARBEIT | MASTER'S THESIS

Titel | Title

Using LP-Modeling to find Exact Solutions for Physician Scheduling Problems with Varying Shift Types and Overlapping Shifts

verfasst von | submitted by

Lisa Maria Nechwatal BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 915
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Betriebswirtschaft
Betreut von | Supervisor:	emer. o. Univ.-Prof. Dipl.-Ing. Dr. Richard Hartl

Contents

Abstract German

Die vorliegende Arbeit befasst sich mit der Problematik der Schichtplanung von Ärzten. In dem hier behandelten Fall besteht der Schichtplan aus unterschiedlichen Schichttypen, welche sich in Wochentags- sowie Wochenendschichten unterscheiden lassen. Zusätzlich zu dieser Unterteilung überlappen sich auch fast alle aneinandergrenzenden Schichten mit einem Zeitraum von mindestens 30 Minuten. Das Datenmaterial und die Eingabedaten wurden von einem Krankenhaus in Deutschland zur Verfügung gestellt, in dem der Schichtplan mit der Zuteilung der Ärzte immer noch manuell erstellt wird. Diese Art der Planung beansprucht einen großen Teil der Arbeitszeit und dieser Zeitaufwand verursacht wiederum hohe Kosten. Beide Faktoren könnten durch die Verwendung eines passenden ganzzahligen linearen Optimierungsmodells verringert beziehungsweise sogar vermieden werden, während zugleich eine optimale Lösung erzielt werden kann. Alle rechtlichen und krankenhausinternen Arbeitsbedingungen, die eingehalten werden müssen, sind in dem bereitgestellten Datenmaterial enthalten und sind dementsprechend auch in dem konstruierten Modell berücksichtigt. Die Zielfunktion des Optimierungsproblems ist so formuliert, dass das Bestreben von ganzen freien Wochenenden für Ärzte bestmöglich erreicht wird. Eine genauere Beschreibung dafür ist, dass keine einzige Wochenendschicht in einer Woche zu einem Arzt zugeteilt werden soll, wenn dieser Arzt am Wochenende dieser Woche frei hat. Abstrakt gesehen wird unterschieden ob ein Arzt an einem Wochenende arbeiten muss oder nicht und dies kann somit binär dargestellt werden. Zusätzlich zu dieser Optimierung wird in

einem zweiten Modell die gleichmäßige Verteilung der Arbeitsbelastung, welche einen weiteren möglichen Aspekt zur Optimierung der Schichtzuteilung darstellt, in Betracht gezogen. Dieses zweite Ziel wird durch eine Erweiterung der Zielfunktion und einer zusätzlich hinzugefügten Bedingung in die bestehende Problemformulierung integriert. Das Ziel dieser ausgebauten Formulierung ist es aufzuzeigen, wie durch eine kleine Ergänzung des Modells, die Verteilung der zugeteilten Schichten über den gesamten Planungshorizont deutlich fairer gestaltet werden kann. Dabei wird die Laufzeit des Programms, durch diese Änderung, nicht wesentlich beeinträchtigt. Mithilfe der Modellierungssprache FICO XPress Mosel ist die Problemstellung als ganzzahlige lineare Optimierung formuliert beziehungsweise modelliert. Dieses Modell wird anschließend mit dem FICO XPress Optimizer gelöst. Eine Lösung wird sowohl für die erste Zielsetzung als auch für die zweite genannte Zielsetzung generiert und individuell besprochen. Weiters folgt auch ein Vergleich der resultierenden Schichtpläne um die erzielbare Verbesserungsmöglichkeit der fairen Schichtenaufteilung zu verdeutlichen. Die erzielten Ergebnisse zeigen auf wie schnell ganzzahlige lineare Optimierung einen Schichtplan für Planungsprobleme mit überlappenden Schichten generieren kann, im Vergleich zu manuellen Lösungswegen. Zusätzlich bietet diese Lösungsmethode auch ein Ausmaß an fairen beziehungsweise ausgeglichenen Verteilungen der Arbeitslast, welche händisch zu diesem Grad wohl kaum zu erreichen wären. Durch die Lösbarkeit kann auch erwähnt werden, dass die Nutzung einer ganzzahligen linearen Optimierung auch eine wesentliche Kostenersparnis bedeuten kann, da weniger menschliche Ressourcen für die Planung benötigt werden. Abschließend wird eine Sensitivitätsanalyse durchgeführt, um feststellen zu können in welchen Bereichen Engpässe zu finden sind und wie jene eventuell umgangen werden können. Ein weiteres Ziel dieser Analyse ist es, herauszufinden wie die Leistungsfähigkeit des Modells durch die Änderung von einzelnen Parametern beeinträchtigt wird. Der geänderte Parameter wurde vor jeder weiteren Änderung wieder zurückgesetzt um jede Analyse auf Basis der ursprünglichen Daten durchführen zu können.

Abstract English

This research addresses a physician scheduling problem which involves a shift plan where weekends and weekdays consist of different overlapping shift types. The input data was provided by a hospital in Germany and concerns a real world problem where the scheduling is still performed manually causing great expenses of time and costs. This data input covers the legal and institutional constraints that have to be satisfied. The objective value is formulated around the goal to create proper weekends for the physicians, which means that they are not assigned to a weekend shift at all on their weekend off. Besides that, fairness is introduced to showcase how a simple additional constraint and objective extension can offer a much more fair allocation of the shifts throughout the planning horizon. Thereby the runtime is not affected in a meaningful manner. FICO Xpress Optimizer is used to solve the problem which is formulated as an integer program with the help of FICO XPress Mosel. The obtained results show how fast and fair a schedule for overlapping shifts can be created using integer programming compared to the current practice of doing it manually. Further, a sensitivity analysis is done to see where bottlenecks lie and how the performance of the model is affected by changing parameters.

1 Introduction

Scheduling is used in various different use sectors, no matter if the entities planned are from a manufacturing sector or from a service sector (Gunawan and Lau, 2013). The manufacturing sector utilizes scheduling in the form of vehicle scheduling, machine scheduling, vendor scheduling or also personnel scheduling whereas the service sector mainly is in need of personnel scheduling but also facility scheduling (Nahmias, 2009). Overall, one of the biggest distinctions between the scheduling variants of the two sectors can be drawn from the objectives they have. Whilst the scheduling goal in a manufacturing environment focuses mainly on the minimization of total costs it also evolves around the satisfaction of the preferences of the people scheduled in the service field. Thus, the service sector is often more complicated to solve as it takes conflicting objectives into account or is supposed to find a good trade off (Gunawan and Lau, 2013).

Nowadays creating high quality staff schedules is one of the most important factors within all service sectors in order to stay profitable and also achieve satisfaction throughout the customer as well as the employee sector (Aykin, 2000). In the health care sector and particularly in a hospital setting it is especially important to generate schedules or rosters that are taking into account the preferences of nurses and physicians, as their contentment is of utmost importance to maintain staff levels up high. This need is most likely due to that it can be very hard and also costly to find someone filling the vacant position (Villarreal and Keskinocak, 2016), (Aiken, 2002). Another difficulty that this sector is facing are a lot of challenges when it comes to creating a schedule and needs to consider a lot of constraints whilst also focusing on covering all shifts with the required amount of human resources. Overall, the physician scheduling problem has not received too much attention when compared to other personnel or employee scheduling problems (Bruni and Detti, 2014), (Cildoz et al., 2021).

In particular the physician scheduling problem is about finding a feasible and preferably optimal work schedule which respects multiple constraints given. The most important constraint being the fulfillment of the requirement to have the proper number of physicians needed assigned for each shift and in addition being as fair as possible. Further, the preferences of the physicians should be taken into account. A schedule of this type can be created for rather short periods starting from two weeks or also for multiple months up to half a year. This naturally depends on the preferences and needs of the department concerned (Gendreau et al., 2007), (Erhard et al., 2018). Overall, it is a procedure done in a single step that needs to find a schedule which covers the given demand and also minimizes the defiance of the scheduling rules determined (Carter and Lapierre, 2001).

Finding a solution for a physician scheduling problem by hand is very time consuming and it is also almost impossible to achieve optimality when considering the multitude of constraints. In specific cases of overlapping shifts and different shift types, the complexity increases even more. The complexity then again can vary depending if all days are treated equally or if weekends are treated differently than the days from Monday through Friday. Further, there is the need for days off between the assigned shifts. The amount of constraints increases with each additional physician considered and thus the creation of a feasible schedule gets severely difficult (Gendreau et al., 2007).

When it comes to solving scheduling problems overall, three types of methods can be stated for categorization: exact methods, methods leading to potential optimal solutions as heuristics and hybrid solutions. Exact methods lead to guaranteed optimal solutions and can be for example achieved by integer programming (IP) or constraint programming (CP). Heuristics do not necessarily lead to optimal solutions, but are able to achieve very good solutions with high quality and use a more reasonable and speedier computation time than when an exact method is used. The use of a combination of integer programming and metaheuristics has recently become very common in the literature as it allows for a more flexible and stronger approach (Turhan and Bilgen, 2020).

Linear programs do lead to guaranteed optimal solutions and depending on the size of the instance also within a reasonable runtime. As the size of the workforce of physicians within a department that a schedule is made for usually is not excessively large. A working LP can safe lots of time for setting up a schedule while also delivering optimality and fairness. Further, a base code program can be adjusted for other departments and also the time saved from the reoccurring need to create a feasible schedule is going to pay off the time used to write a working program rather quickly (Ernst et al., 2004).

It is important to note that the staffing requirements can vary a lot between departments but also between the same departments between different countries. According to the WHO these variances occur due to differing accessibility of hospitals, varying morbidity rates and of course economic circumstances in different regions. Yet there are ratio methods for staffing in different departments which means that the realm of how much staff is used in the same type of department in different hospitals is fairly equal in per cent of the total hospital workforce (Geneva:

World Health Organization, 2023).

Overall, when browsing the literature it is very easy to see that there is almost none, if not none at all, covering overlapping shifts. In addition, the variation in shift types used is very often limited and reduced to the shifts having the same durations as well as weekends being structured the same as weekdays. Therefore, this paper can help to bridge a gap of rarely considered aspects and constraints in mathematical programming for the physician scheduling problem.

2 Literature Review

There is several literature regarding shift scheduling problems. Therefore, the focus of the review is on work that complies with employee shift scheduling in the healthcare sector and some more general approaches for employee scheduling. A majority of literature surrounding medical staff scheduling focuses on metaheuristics and its applications in combination with linear programming like Legrain et al. (2015), Turhan and Bilgen (2020), Cildoz et al. (2021) or Adhiana et al. (2023) and not too many cover linear programming or constraint programming on its own like Trilling et al. (2006), Jaumard et al. (1998) and Bruni and Detti (2014).

The bigger portion of recently released literature concerning the physician scheduling problem, like Cildoz et al. (2021), focuses on hybrid methods that consist of a combination of exact methods and metaheuristics. This can be explained by often unreasonable runtimes of exact methods for rather large instances. Thus, using heuristics with destroy methods can help to create a smaller problem. This then is used for LP-modeling and after that repair methods are applied, which add constraints back to the solutions gained. These methods can help to control on how to intensify and also diversify the exploration of high quality solutions within the search space (Mansini and Zanotti, 2020).

The physician scheduling problem is related to various other scheduling problems, especially the nurse scheduling problem as it also operates in a hospital setting with similar constraints as can be seen from the following description of the similarities and distinctions to regular employee scheduling problems. However, there are certain factors that make each problem unique (Gunawan and Lau, 2013).

At first the nurse scheduling problem, like the physician scheduling problem, seems to be quite similar to a regular employee shift scheduling problem, but it is more complex and it is harder to achieve an optimal solution. This is due the fact that hospitals need to be staffed for 24 hours every single day and a majority of legal constraints need to be respected. Then there is also the goal to create a schedule that takes fairness into account. Overall, there is a multitude of hard and soft constraints that are specifically required when scheduling shifts in a hospital (Cheang et al., 2003). In particular, the satisfaction of the physicians is seen as one of the most essential aspects, as it is of utmost importance to retain physicians within a hospital. Further, physician scheduling does not need to adhere to as many legal constraints, and thus is also able to be lead by more personal preferences and is more compliant compared to nurse rostering (Gunawan and Lau, 2013).

The focus of the work scope mainly is on one single duty only when physician scheduling problems are described and conquered. Research recorded in literature papers like Carter and Lapierre (2001), Beaulieu et al. (2000), Ogulata et al. (2008) or Vanberkel et al. (2011) focus commonly on scheduling for emergency rooms, more rarely on surgery rooms and seldom on facilities like rehabilitation services. Concluding it can be seen that the focal point of research in physician scheduling lies on one assigned duty at once. Thus Gunawan and Lau (2013) took upon the task to take the whole scope of daily tasks into account and conduct research regarding that.

The recent literature mainly concentrates on hybrid approaches that then are tested for different instances like Legrain et al. (2015), Turhan and Bilgen (2020), Santos et al. (2016), Adhiana et al. (2023) and others. This leads to a minority of case specific approaches that then are also rather directly usable in real world cases. Thus the approach of this paper is supposed to offer a rather cheap to almost free improvement for smaller hospital departments that are able to save enormous time which currently still is spent on creating a schedule by hand.

A short overview containing examples of literature which discusses scheduling problems solved using purely mathematical optimization can be seen in Table 2.1. Further an overview of literature that uses various types of metaheuristics to solve scheduling problems is given in Table 2.2.

Authors	Year	Method	Type	Speciality
Bruni et al.	2014	MIP	PSP	Exact solution possible using Branch and Cut
Adhiana et al.	2023	IP	ESS	Small number of constraints and optimal solution
Brenner et al.	2009	MIP	PSP	No given shifts - shifts are created for the problem
Bakr et al.	2023	CP	PSP	Comparison regarding hard constraints and additional soft constraints
Marchesi et al.	2020	SP	PSP	Performance evaluation using stochastic discrete-event simulation (2nd Stage)

Table 2.1: Approaches using only mathematical optimization (Single-Stage Scheduling Problem)

Authors	Year	Method	Type	Speciality
Turhan et al.	2020	MIP & SA	NRP	Initial relaxed solution created by various exact methods
Childoz et al.	2021	IP, GRASP,VND & NFO	PSP	Uses real work calendar and one year planning horizon
Mansini et al.	2020	MILP & ALNS	PSP	Allocation focus of tasks and shift times
Gunawan et al.	2013	MP & LS	PSP	Distinction between small and large instances

Table 2.2: Approaches using a combination of mathematical optimization and metaheuristics (Multi-Stage Scheduling Problem)

If you look for overlaps in shift scheduling problems in the existing literature, there are two papers that stood out and do both offer an approach to handle some sort or form of overlapping constraint. Addou and Soumis (2007) cover a problem for overlapping breaks within the same shift type and make use of the Bechthold-Jacobs generalized shift-scheduling problem and extend it with further formulations to allow for, what they call, 'extraordinary overlap'. Further, they offer a comparison and an insight into other extensions which have been performed. Kletzander and Musliu (2020) made an effort to form generalized versions of various constraints for different kinds of job scheduling problems. Therefore, some sort of toolkit is formed that can be used to pick the needed constraints to create models for various scheduling problems. Under these generalized formulations, a constraint to handle overlaps is also covered in the form of a heuristic. Concluding, these two approaches do not fit the need that is given by the instance that this paper is concerned with.

3 Method Description

3.1 Term Definition

In the following paragraph essential and necessary term definitions for the physician scheduling problem are provided to ensure the clarity of the method description.

- **Physician:** A person who is a medical professional who is trained and licensed to practice medicine.
- **Shift:** A time period during which physicians perform their duties. There are multiple shifts in a day (i.e. early shift, morning shift, late shift, night shift) and they are pre-defined regarding their start and end times and therefore also their duration.
- **Schedule:** A timetable which shows the assigned physicians to each shift on each day.
- **Planning horizon:** Length of one schedule. Each planning horizon is given in advance.
- **Constraints:** Constraints are defined by laws and regulations. For example, there is a mandated minimum period of rest following a shift. Thus, a shift cannot be assigned to a physician that starts within less than the determined resting time after the end of the previous shift.

3.2 Model Outline

The physician scheduling problem discussed in this paper is based on a real world case from a hospital in Germany where the scheduling is still done manually. The data used to create the code for this model has been provided by that hospital. Generally, the goal is that the model should substitute the current need for scheduling manually. The physicians in this case are all treated equally, which means that there are no seniority levels that need to be considered in the planning and they all underlie the constraints of the problem to the same extent.
In order to introduce the outlines of the model in a versatile manner, it has been decided to co-note the data input as parameters. This generalization of the model should enable the formulation to be very logical and easy to understand how it can be adapted for other cases.

3.3 Model Data & Constraints

Before stating the constraints needed for the model, it is necessary to clarify the main given shift types and their relation to the different days of the week that the mathematical model has been formed around.

3.3.1 Shift Types

The schedule created by this model spans over a planning horizon of twelve weeks. Each week consists of seven working days (Monday to Sunday) and each day consists of several shifts depending on if it is a weekday or a weekend-day. Weekdays range from Monday through Friday and consist of four shifts each, namely early shift, morning shift, late shift and night shift. Weekends only concern Saturdays and Sundays and consist of three shifts each, namely morning shift, day shift and night shift. An overview of all this data can be seen in Table 3.1. Even though Friday is considered a weekday, the last two shifts on that day, namely the late shift and the night shift, count as working on weekends in addition to shifts on Saturdays and Sundays. The starting and ending time of each shift type as well as the corresponding duration can be seen in table 3.2 for the weekday shifts and in table 3.3 for the weekend shifts.

No. Physicians	No. Weeks	No. Days/Week	No. Shift Types Total	No. Shift Types Weekdays	No. Shift Types Weekend
8	12	7	7	4	3

Table 3.1: Input data of real world scenario

(1) Early Shift	(2) Morning Shift	(3) Late Shift	(4) Night Shift
07:00-15:00	10:00-19:30	15:00-23:00	23:00-07:30
8 hours	9.5 hours	8 hours	8.5 hours

Table 3.2: Starting and ending times of the weekday shift types

(5) Morning Shift	(6) Day Shift	(7) Night Shift
07:00-19:30	13:00-21:30	19:00-07:30
12.5 hours	8.5 hours	12.5 hours

Table 3.3: Starting and ending times of the weekend shift types

A visualization of the different shift types and the allocation to which day of the week they belong to can be seen in figure 3.1. There is an overlap with a duration of at least 30 minutes over every single adjacent shift except for one pair. The pair of shifts which does not create an intersection between their ending and starting times are the late shift and the night shift of the weekdays. Further, there is a version in figure 3.2 with a tighter time frame that covers the shifts from Friday to the beginning of Monday where the overlaps are illustrated in a more visible way.

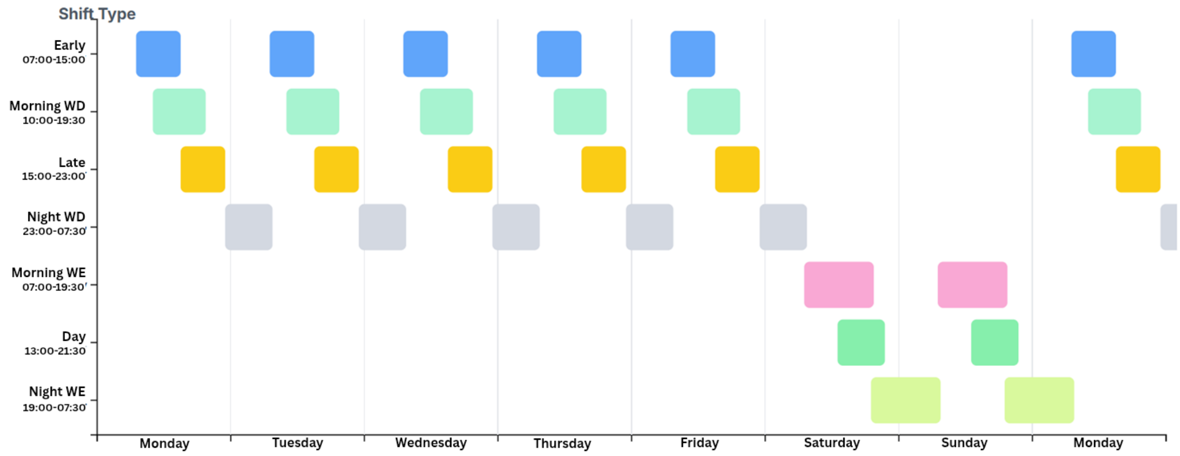

Figure 3.1: Overview of different shift types

3.3.2 Constraints

1. Exactly one physician has to be assigned to each shift.
2. Each physician can work a maximum of 42.5 hours per week.
3. After each shift a rest period of at least ten hours has to follow.
4. A maximum of ten consecutive working days are allowed and if this threshold is reached, at least the following two days need to be days off.
5. A physician is not allowed to work more than four night shifts in a row.
6. Each physician is allowed to work on a maximum of two weekends per month (within four weeks).
7. The corresponding amount of worked days on a weekend need to be given off in the form of weekday shifts during the week (Monday - Friday) within the following two weeks.

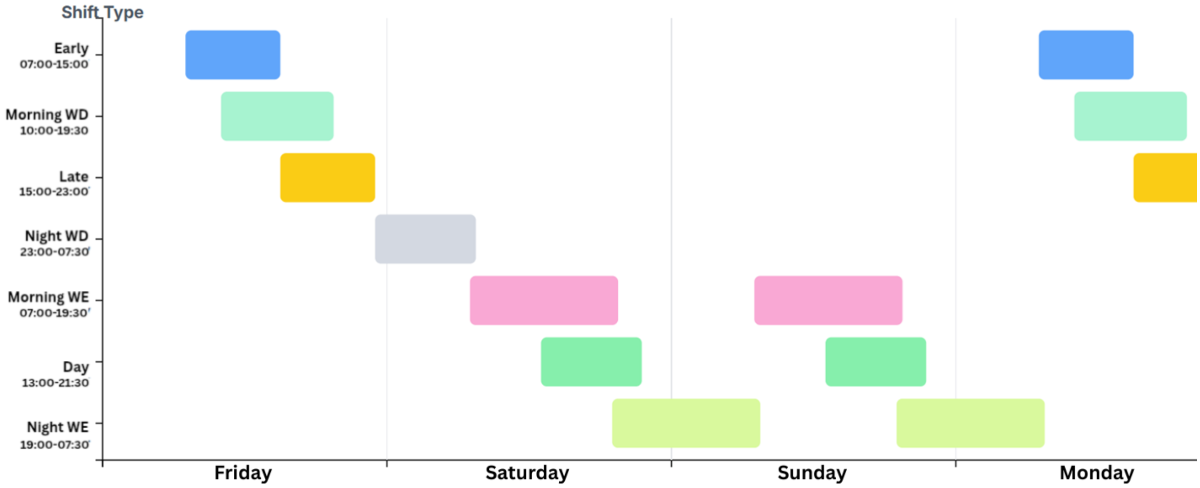

Figure 3.2: Overview of overlapping extent

3.4 First Objective - Weekend-Work

The first objective is formulated to minimize weekend-work. Weekend-work is understood as working at least one of the shifts belonging to the weekend. E.g. if a physician is scheduled for the night shift on a Friday it is preferred that this person is also scheduled for another shift on Saturday or Sunday instead of just working one shift belonging to the weekend. This means that the count for working on a weekend does not depend on the number of shifts assigned, but once a single weekend shift is assigned, the physician is considered to work on that weekend. This results in physicians to have proper weekends off and have a longer rest period. Additionally their off days will not be split apart in each week and they can make more use of their leisure time. For the following content, it is important to note, that the model formulated for this first objective does not take a fair allocation of the shifts into account.

3.5 Mathematical Formulation

3.5.1 Data Declarations

Set of...

$P\ldots$ physicians

$W\ldots$ weeks

$D\ldots$ days

$D_1\ldots$ weekdays

$D_2\ldots$ weekend days

$S_1\ldots$ shift types occurring on weekdays

$S_2\ldots$ shift types occurring on weekend days

$S\ldots$ $\mathrm{S}_1 \cup \mathrm{S}_2$

Array of the...

$BG\_S\ldots$ start times of each shift $s \in \mathrm{S}_1$

$END\_S\ldots$ end times of each shift $s \in \mathrm{S}_1$

$BG\_WE\_S\ldots$ start times of each shift $s \in \mathrm{S}_2$

$END\_WE\_S\ldots$ end times of each shift $s \in \mathrm{S}_2$

$SDURATION\ldots$ durations of each shift type $s \in \mathrm{S}$

Parameter that states...

$CONS\_WE\_S\ldots$ how many shifts of the week are considered for weekend work

$DAYSOFF\ldots$ the minimum number of days off after MAX_CONS_DAYS consecutive days

$REG\_F\_S\ldots$ how many shifts on a Friday count as regular shifts

$MAX\_CONS\_DAYS\ldots$ how many consecutively scheduled days are allowed

$MIN\_BREAK\ldots$ how many hours need to be between two scheduled shifts

$MAX\_HOURS\ldots$ how many hours a physician is allowed to work in a week

$MAXNS\ldots$ the maximum number of consecutive night shifts allowed within a determined time horizon (here detected as MAX_WEW_W)

$MAX\_WEW\_W\ldots$ the time horizon (in weeks) considered for a determined maximum amount of weekend work (here detected as MAX_WEW)

$MAX\_WEW\ldots$ the maximum amount of weekend work allowed within the determined time horizon (in weeks) (here detected as MAX_WEW_W)

3.5.2 Decision Variables

assign(p, w, s, d) ... decision variable that states if physician $p \in \mathrm{P}$ in week $w \in \mathrm{W}$ is assigned to shift $s \in \mathrm{S}$ on day $d \in \mathrm{D}$

weekend_work(p, w) ... decision variable that states if physician $p \in \mathrm{P}$ works on a weekend shift in week $w \in \mathrm{W}$

work_on_day(p, w, d) ... decision variable that states if physician $p \in \mathrm{P}$ works on day $d \in \mathrm{D}$ in week $w \in \mathrm{W}$

work_on_friday_regular(p, w) ... decision variable that states if physician $p \in \mathrm{P}$ works an early or morning shift on a Friday in week $w \in \mathrm{W}$

nightshift_work(p, w, d) ... decision variable that states if physician $p \in \mathrm{P}$ works a nightshift on day $d \in \mathrm{D}$ in week $w \in \mathrm{W}$

consecutive_days(d, w, day) ... decision variable that states how many days in a row physician $p \in \mathrm{P}$ worked by day $d \in \mathrm{D}$ in week $w \in \mathrm{W}$

max_days(p, w, d) ... decision variable that states if physician $p \in \mathrm{P}$ has worked _MAX_CONS_DAYS days in a row by day $d \in \mathrm{D}$ in week $w \in \mathrm{W}$

cons_days_endpweek(p, w) ... decision variable that states how many consecutive days physician $p \in \mathrm{P}$ has worked by the last day of week $w \in \mathrm{W}$

3.5.3 Domain Specification of Decision Variables

assign$(p, s, w, d) \in \{0, 1\}$	$\forall p \in P, s \in S, w \in W, d \in D$
work_on_day$(p, w, d) \in \{0, 1\}$	$\forall p \in P, w \in W, d \in D$
nightshift_work$(p, w, d) \in \{0, 1\}$	$\forall p \in P, w \in W, d \in D$
work_on_friday_regular$(p, w) \in \{0, 1\}$	$\forall p \in P, w \in W$
weekend_work$(p, w) \in \{0, 1\}$	$\forall p \in P, w \in W$
consecutive_days$(p, w, d) \geq 0$	$\forall p \in P, w \in W, d \in D$
max_days$(p, w, d) \in \{0, 1\}$	$\forall p \in P, w \in W, d \in D$
cons_days_endpweek$(p, w, d) \geq 0$	$\forall p \in P, w \in W$

3.5.4 Mathematical Formulation

Objective Function: Minimize the sum of the total weekend shifts worked of all physicians $p \in \mathrm{P}$ over all weeks $w \in \mathrm{W}$

$$\min \quad Z = \sum_{p \in P, w \in W} \text{weekend\_work}(p, w) \tag{1}$$

One Physician Per Shift: Exactly one physician $p \in \mathrm{P}$ needs to be assigned to each shift $s \in \mathrm{S}$ on every day $d \in \mathrm{D}$ of each week $w \in \mathrm{W}$

$$\sum_{p \in P} \text{assign}(p, s, w, d) = 1 \qquad \forall w \in W, \forall d \in D_1, \forall s \in S_1 \tag{2}$$

$$\sum_{p \in P} \text{assign}(p, s, w, d) = 1 \qquad \forall w \in W, \forall d \in D_2, \forall s \in S_2 \tag{3}$$

$$\sum_{p \in P} \text{assign}(p, s, w, d) = 0 \qquad \forall w \in W, \forall d \in D_1, \forall s \in S_2 \tag{4}$$

$$\sum_{p \in P} \text{assign}(p, s, w, d) = 0 \qquad \forall w \in W, \forall d \in D_2, \forall s \in S_1 \tag{5}$$

Maximum Weekly Hours: The hours worked per physician $p \in \mathrm{P}$ per week $w \in \mathrm{W}$ must not exceed MAX_HOURS hours

$$\sum_{d \in D} \sum_{s \in S} \text{assign}(p, s, w, d) \cdot \text{SDURATION}(s) \leq \text{MAX\_HOURS}, \tag{6}$$

$$\forall p \in P, \forall w \in W$$

Connecting Decision Variables:

A physician $p \in \mathrm{P}$ is working on a day $d \in \mathrm{D}$, if they are assigned to any shift $s \in \mathrm{S}$ on that day.

$$\text{work\_on\_day}(p, w, d) \geq \frac{\sum_{s \in S} \text{assign}(p, s, w, d)}{|S|}, \quad \forall p \in P, \forall w \in W, \forall d \in D \tag{7}$$

A physician $p \in \mathrm{P}$ is assigned to an early or a morning shift on a Friday, if they are working a shift that does not belong to the weekend.

$$\text{work\_on\_friday\_regular}(p, w) \geq \frac{\sum_{s=1}^{2} \text{assign}(p, s, w, 5)}{\text{REG\_F\_S}}, \quad \forall p \in P, \forall w \in W \tag{8}$$

A physician $p \in \mathrm{P}$ is working on the weekend in a week $w \in \mathrm{W}$, if they are assigned to a shift $s \in \mathrm{S}_2$ on Saturday or Sunday in that week.

$$\text{weekend\_work}(p, w) \geq \frac{\sum_{d=6}^{7} \sum_{s \in S_2} \text{assign}(p, s, w, d) + \sum_{s=3}^{4} \text{assign}(p, s, w, 5)}{\text{CONS\_WE\_S}} \tag{9}$$

$$\forall p \in P, \forall w \in W$$

A physician $p \in \mathrm{P}$ is working a night shift on day $d \in \mathrm{D}$, if they are assigned to a night shift on that day.

$$\text{nightshift\_work}(p, w, d) = \text{assign}(p, 4, w, d) + \text{assign}(p, 7, w, d) \tag{10}$$

$$\forall p \in P, \forall w \in W, \forall d \in D$$

Maximum Weekends Assigned: Ensure only MAX_WEW weekends per MAX_WEW_W weeks are assigned per physician.

$$\sum_{w=(\text{MAX\_WEW\_W}\cdot k+1)}^{\min(\text{MAX\_WEW\_W}\cdot k+\text{MAX\_WEW\_W},\,|W|)} \text{weekend\_work}_{p,w} \leq \text{MAX\_WEW}, \tag{11}$$

$$\forall p \in P, \forall k \in \left\{0, 1, \ldots, \left\lfloor \frac{|W|}{\text{MAX\_WEW\_W}} \right\rfloor - x \right\}$$

Where x is:

$$x = \begin{cases} 0 & \text{if } |W| \bmod \text{MAX\_WEW\_W} = 0 \\ 1 & \text{otherwise} \end{cases}$$

Weekend Shift Reimbursement: Ensure a physician $p \in \mathrm{P}$ gets the worked weekend shifts in week $w \in \mathrm{W}$ off during the following 2 weeks (in regular shifts)

$$\begin{aligned} &\sum_{d=1}^{4}(1 - \text{work\_on\_day}(p, w+1, d)) \\ &+ \sum_{d=1}^{4}(1 - \text{work\_on\_day}(p, w+2, d)) \\ &+ (1 - \text{work\_on\_friday\_regular}(p, w+1)) \\ &+ (1 - \text{work\_on\_friday\_regular}(p, w+2)) \\ \geq &\sum_{d=6}^{7}\sum_{s\in S_2} \text{assign}(p, s, w, d) + \sum_{s=3}^{4} \text{assign}(p, s, w, 5) \end{aligned} \tag{12}$$

$$\forall p \in P, \forall w \in \{1, \ldots, (|W| - 2)\}$$

Maximum Consecutive Night Shifts: Ensure a physician $p \in \mathrm{P}$ works at most MAXNS days in a row, how this is achieved depends on day $d \in \mathrm{D}$

Parameter i results from:

$$i = d + MAXNS$$

If $i \leq 7$ then:

$$\sum_{c=d}^{d+MAXNS} \text{nightshift\_work}(p, w, c) \leq \text{MAXNS} \tag{13}$$

If $i > 7$ and w < |W| then:

Parameter h results from:

$$h = 7 - \text{day}$$

$$\sum_{c=d}^{7} \text{nightshift\_work}(p, w, c) + \sum_{c=1}^{MAXNS-h} \text{nightshift\_work}(p, w+1, c) \leq \text{MAXNS}, \tag{14}$$

$$\forall p \in P, w \in W, d \in D$$

MIN_BREAK-Hour Off Constraint: Ensure there are MIN_BREAK hours off between two assigned shifts of one physician $p \in \mathrm{P}$. This constraint is case distinctive for Friday, Saturday, and Sunday as they interfere with different shift-type groups (weekend or regular) on the respective consecutive day.

Only One Assigned Shift Per Day: Ensure there is a maximum of one assigned shift $s \in \mathrm{S}$ per physician $p \in \mathrm{P}$ on each day $d \in \mathrm{D}$ in each week $w \in \mathrm{W}$.

$$\sum_{s \in S_1} \text{assign}(p, s, w, d) \leq 1 \quad \forall w \in W, \forall d \in D_1, \forall p \in P \tag{15}$$

$$\sum_{s \in S_2} \text{assign}(p, s, w, d) \leq 1 \quad \forall w \in W, \forall d \in D_2, \forall p \in P \tag{16}$$

Mondays through Thursdays (1...4):

- **Same shift type ($s_1 = s_2$):**

$$\text{if } s_1 = s_2 \text{ then}$$

$$\text{assign}(p, s_1, w, d) + \text{assign}(p, s_2, w, d+1) \leq 2, \tag{17}$$

$$\forall p \in P, \forall w \in W, \forall d \in \{1, 2, 3, 4\}, \forall s_1, s_2 \in S_1$$

- **Different shift types, less than MIN_BREAK hours off:**

$$\text{if BEG\_S}(s_2) + 24 - \text{END\_S}(s_1) < \text{MIN\_BREAK then}$$

$$\text{assign}(p, s_1, w, d) + \text{assign}(p, s_2, w, d+1) \leq 1, \tag{18}$$

$$\forall p \in P, \forall w \in W, \forall d \in \{1, 2, 3, 4\}, \forall s_1, s_2 \in S_1$$

Fridays (5): (Shift s_1 is a regular shift and shift s_2 is a weekend shift)

$$\text{if BEG\_WE\_S}(s_2) + 24 - \text{END\_S}(s_1) < \text{MIN\_BREAK then}$$

$$\text{assign}(p, s_1, w, 5) + \text{assign}(p, s_2, w, 6) \leq 1, \tag{19}$$

$$\forall p \in P, \forall w \in W, \forall s_1 \in S_1, \forall s_2 \in S_2$$

Saturdays (6):

- **Same shift type ($s_1 = s_2$):**

$$\text{if } s_1 = s_2 \text{ then}$$

$$\text{assign}(p, s_1, w, 6) + \text{assign}(p, s_2, w, 7) \leq 2, \tag{20}$$

$$\forall p \in P, \forall w \in W, \forall s_1, s_2 \in S_2$$

- **Different shift types, less than MIN_BREAK hours off:**

$$\text{if BEG\_WE\_S}(s_2) + 24 - \text{END\_WE\_S}(s_1) < \text{MIN\_BREAK then}$$

$$\text{assign}(p, s_1, w, 6) + \text{assign}(p, s_2, w, 7) \leq 1, \tag{21}$$

$$\forall p \in P, \forall w \in W, \forall s_1, s_2 \in S_2$$

Sundays (7): (Shift s_1 is a weekend shift and shift s_2 is a regular shift)

$$\text{if BEG\_S}(s_2) + 24 - \text{END\_WE\_S}(s_1) < \text{MIN\_BREAK then}$$

$$\text{assign}(p, s_1, w, 7) + \text{assign}(p, s_2, w+1, 1) \leq 1, \tag{22}$$

$$\forall p \in P, \forall w \in \{1, \ldots, (|W| - 1)\}, \forall s_1 \in S_2, \forall s_2 \in S_1$$

MAX_CONS_DAYS Consecutive Workdays, DAY_OFF Days Off Constraint: After MAX_CONS_DAYS days of consecutive work, DAYSOFF days off need to be scheduled.

Relationship between consecutive days and working on a day:

$$\text{consecutive\_days}(p, w, d) \geq \text{work\_on\_day}(p, w, d), \tag{23}$$

$$\forall p \in P, \forall w \in W, \forall d \in D$$

Consecutive days at the end of the previous week:

$$\text{cons\_days\_endpweek}(p, w) = \text{consecutive\_days}(p, w - 1, 7), \tag{24}$$

$$\forall p \in P, \forall w \in \{2, \ldots, |W|\}$$

First day of the first week:

$$\text{consecutive\_days}(p, 1, 1) = \text{work\_on\_day}(p, 1, 1), \tag{25}$$

$$\forall p \in P$$

First day of the subsequent weeks (upper bound):

$$\text{consecutive\_days}(p, w, 1) \leq \text{cons\_days\_endpweek}(p, w) + work\_on\_day(p, w, 1), \tag{26}$$

$$\forall p \in P, \forall w \in \{2, \dots, |W|\}$$

First day of the subsequent weeks (lower bound):

$$\begin{aligned} \text{consecutive\_days}(p, w, 1) &\geq \text{cons\_days\_endpweek}(p, w) + \text{work\_on\_day}(p, w, 1) \\ &- (1 - \text{work\_on\_day}(p, w - 1, 7)) \cdot \text{MAX\_CONS\_DAYS}, \end{aligned} \tag{27}$$

$$\forall p \in P, \forall w \in \{2, \dots, |W|\}$$

Maximum consecutive days on the first day of the subsequent weeks:

$$\text{consecutive\_days}(p, w, 1) \leq \text{MAX\_CONS\_DAYS} \cdot \text{work\_on\_day}(p, w, 1), \tag{28}$$

$$\forall p \in P, \forall w \in \{2, \dots, |W|\}$$

Calculating consecutive days within the same week (upper bound):

$$\text{consecutive\_days}(p, w, d) \leq \text{consecutive\_days}(p, w, d - 1) + 1, \tag{29}$$

$$\forall p \in P, \forall w \in W, \forall d \in \{2, \dots, |D|\}$$

Calculating consecutive days within the same week (lower bound):

$$\text{consecutive\_days}(p, w, d) \geq \text{consecutive\_days}(p, w, d-1) + 1 - (1 - \text{work\_on\_day}(p, w, d-1)) \cdot \text{MAX\_CONS\_DAYS}, \tag{30}$$

$$\forall p \in P, \forall w \in W, \forall d \in \{2, \ldots, |D|\}$$

Checking for MAX_CONS_DAYS consecutive days (lower bound):

$$\text{max\_days}(p, w, d) \geq \frac{\text{consecutive\_days}(p, w, d) - (\text{MAX\_CONS\_DAYS} - 1)}{1}, \tag{31}$$

$$\forall p \in P, \forall w \in W, \forall d \in D$$

Checking for MAX_CONS_DAYS consecutive days (upper bound):

$$\text{max\_days}(p, w, d) \leq \frac{\text{consecutive\_days}(p, w, d)}{\text{MAX\_CONS\_DAYS}}, \tag{32}$$

$$\forall p \in P, \forall w \in W, \forall d \in D$$

Minimum Day Off Constraint: DAYSOFF days off after MAX_CONS_DAYS consecutive days.

Parameter e results from:

$$e = d + \text{DAYSOFF}$$

If $e < 7$ then:

$$\sum_{c=d+1}^{e} \text{work\_on\_day}(p, w, c) \leq \text{DAYSOFF} \cdot (1 - \text{max\_days}(p, w, d)) \tag{33}$$

If $e = 7$ and w < |W| then:

$$\sum_{c=d+1}^{7} \text{work\_on\_day}(p, w+1, c) \leq \text{DAYSOFF} \cdot (1 - \text{max\_days}(p, w, d)) \tag{34}$$

If $e > 7$ and w < |W| then:

$$\sum_{c=d+1}^{\text{DAYSOFF}} \text{work\_on\_day}(p, w, c) + \sum_{c=1}^{f} \text{work\_on\_day}(p, w+1, c) \leq \text{DAYSOFF} \cdot (1 - \text{max\_days}(p, w, d)) \tag{35}$$

$$\forall p \in P, w \in W, d \in D$$

Where f is:

$$f = e - 7$$

3.6 Model Extension

The model (1) - (35) delivers a solution for the first objective of minimizing weekend-work and also assures that exactly one physician is assigned to each shift with a constraint. This creates favorable assignments to cover the weekend shifts, where a physician either has a proper weekend off, which means the shifts contributing to weekend-work are not assigned at all to this physician in that week, or has to work multiple of these weekend-work shifts. This schedule covers the whole planning horizon and respects the required resting periods as well as the maximum work time per week and physician. Of course the maximum streak of assigned days of each physician are adhered to too. Although the solution obtained by this basic model is feasible and also optimal for the set objective too, it is not considering how the workload allocation between all physicians is affected.

As already mentioned above, other factors that are important in a real world setting, like the workload assigned to each physician, are neglected. This aspect of assignment deviation is still rather large up to this point in the result of this model. To account for this, a second objective which takes care of exactly this issue is introduced to the existing objective in a following model. This reformulated objective ensures a fair distribution of the amount of assigned shifts throughout all physicians within the planning horizon and still provides proper weekends. The complete concept of this second objective is described in the following section.

3.7 Second Objective - Fairness

For the model for the second objective, the model and objective made for the first objective are taken over and are extended to provide a fair allocation throughout the entire planning horizon. This fairness aspect is designed to focus on the workload in the form of amount of total shifts and not total hours assigned to each physician within the planning horizon, which means the actual working hours are not considered. To achieve this, the objective function detects the deviation of the amount of the assigned shifts between each possible pair of physicians and minimizes the sum of all these deviations. For this to work an additional constraint as well as two deviation variables are necessary. The constraint is formulated to subtract the sum of assigned shifts of one physician with sum of assigned shifts of the other and that should be as close to zero as possible, therefore the constraint equals zero. For this constraint to always equal zero two variables are necessary as the subtraction could be also negative or positive, and following that either a subtraction or addition of a deviation variable is needed.

Further it is important to note that this fairness is given by the whole planning horizon, therefore it can occur that in one week a physician barely has to work whilst another one has to work the maximum workload possible and in another week it is switched around leading to evening out the amount of allocated shifts.

The adjusted objective function is shown in (36), where the objective function of the model for the first objective is extended by adding the two deviation variables explained earlier. The adjoined constraint (37) showcases where the actual workload comparison takes place. With these extensions the best achievable fairness throughout the schedule and the minimized overall weekend-work, can be found.

This was also done by using an epsilon constraint, where the program generated a Pareto-front and optimized fairness as a second objective. It only took one optimization round and it found the same solution as the combined objective function, which means that the best possible solution was found instantly. Therefore, there is no benefit in using the epsilon constraint over the combined objective function described above. This is possible as both objectives require minimization.

All following sensitivity analyses were conducted using either the model for the second objective or a combination of both models, depending on what aspect is discussed.

Decision Variables

dev_one$_{p,p}$ … decision variable that states how much positive deviation regarding assigned shifts is between physician $p \in \mathrm{P}$ and physician $p \in \mathrm{P}$

dev_two$_{p,p}$ … decision variable that states how much negative deviation regarding assigned shifts is between physician $p \in \mathrm{P}$ and physician $p \in \mathrm{P}$

Domain Specification of Decision Variables

$$\text{dev\_one}_{p,p} \geq 0 \qquad \forall p \in P, p \in P$$

$$\text{dev\_two}_{p,p} \geq 0 \qquad \forall p \in P, p \in P$$

Objective Function: Minimize the sum of the total weekend shifts worked of all physicians $p \in \mathrm{P}$ and the positive and negative deviation between the number of all assigned shifts for all combinations of physicians $p \in \mathrm{P}$

$$\min Z = \sum_{p \in P, w \in W} \text{weekend\_work}(p, w) + \sum_{p_1, p_2 \in P} \text{dev\_one}(p_1, p_2) + \sum_{p_1, p_2 \in P} \text{dev\_two}(p_1, p_2) \tag{36}$$

Fairness Regarding Total Assigned Shifts Between Physicians: For every pair of distinct physicians (p_1, p_2) where $p_1 \in P$ and $p_2 \in P$ with $p_1 \leq p_2$, the difference in the total number of assigned shifts (both regular and weekend) over all weeks must be equal to the difference between their respective deviation variables:

$$\begin{aligned} \sum_{s \in S, w \in W, d \in D} \text{assign}(p_1, s, w, d) - \sum_{s \in S, w \in W, d \in D} \text{assign}(p_2, s, w, d) \\ + \text{dev\_one}(p_1, p_2) - \text{dev\_two}(p_1, p_2) = 0, \end{aligned} \tag{37}$$

$$\forall p_1 \in P, \forall p_2 \in P \mid p_1 \leq p_2$$

3.8 Variability & Complexity

For simplification in this and the following sections the model for the first objective will be referred to as basic model and the model for the second objective will be referred to as extended model.

As the extended model is essentially containing the basic model, the following section refers to the extended model. The model is forming the basis to create a schedule that fulfills all minimum and maximum constraints, for each physician in this setting. At the same time it is ensuring that the model operates in a variable environment, where the data input can be changed without having to change anything in the model formulation itself.

An added difficulty can be found within the iteration through the weeks as the formulation of various constraints require multiple versions for different days of the week. The way of the formulation then depends on the currently relevant day as it can differ in how many days from the past or future week need to be considered to fulfill the corresponding constraint.

4 Case Study: Physician Scheduling

4.1 Solving the Model

The model was coded using the FICO Xpress Mosel language and solved with the FICO XPress Optimizer. All computations were performed on a 1.60 GHz PC (Intel(R) Core(TM) i5-10210U CPU).

Table 4.1 gives an overview of the variables and number of constraints necessary to obtain a solution for this instance for the whole twelve week planning horizon. These numbers apply for both models discussed in the following sections, as both are formulated with variables which serve the purpose of showing deviation values.

Total no. variables	No. binary variables	No. integer variables	No. Rows
5 520	4 760	760	11 856

Table 4.1: Problem Size for a twelve week planning horizon - first objective

The comparisons made are based on the deviations that occur between the assigned shifts of the physicians, which means if a model has an overall higher deviation, it is performing worse than the one with the lower deviation. All deviations are compound by the sum of the deviations between each possible pairing of physicians. To further compare the fairness of the two models in more detail, the biggest deviation between two physicians for the total amount of assigned shifts, assigned weekend-work and also average weekly working hours have been analyzed.

4.2 Solution for the First Objective

The model was able to find an optimal solution within 1.9 seconds. Figure 4.1 displays the schedule for the first week that showcases which physician is assigned to which shift, on which day. The term doctor is used instead of physician for this figure. However, this is only supposed to give an example on how the visualization of a resulting schedule could look like for a weekly overview. A schedule for the whole planning horizon is displayed in figure 4.2. The shift type assigned on a day is represented by the numbers in the cell. Which number is corresponding to which shift type is defined in tables 3.2 and 3.3. Cells highlighted in grey indicate assigned weekday shifts whilst cells highlighted in orange indicate assigned weekend shifts.

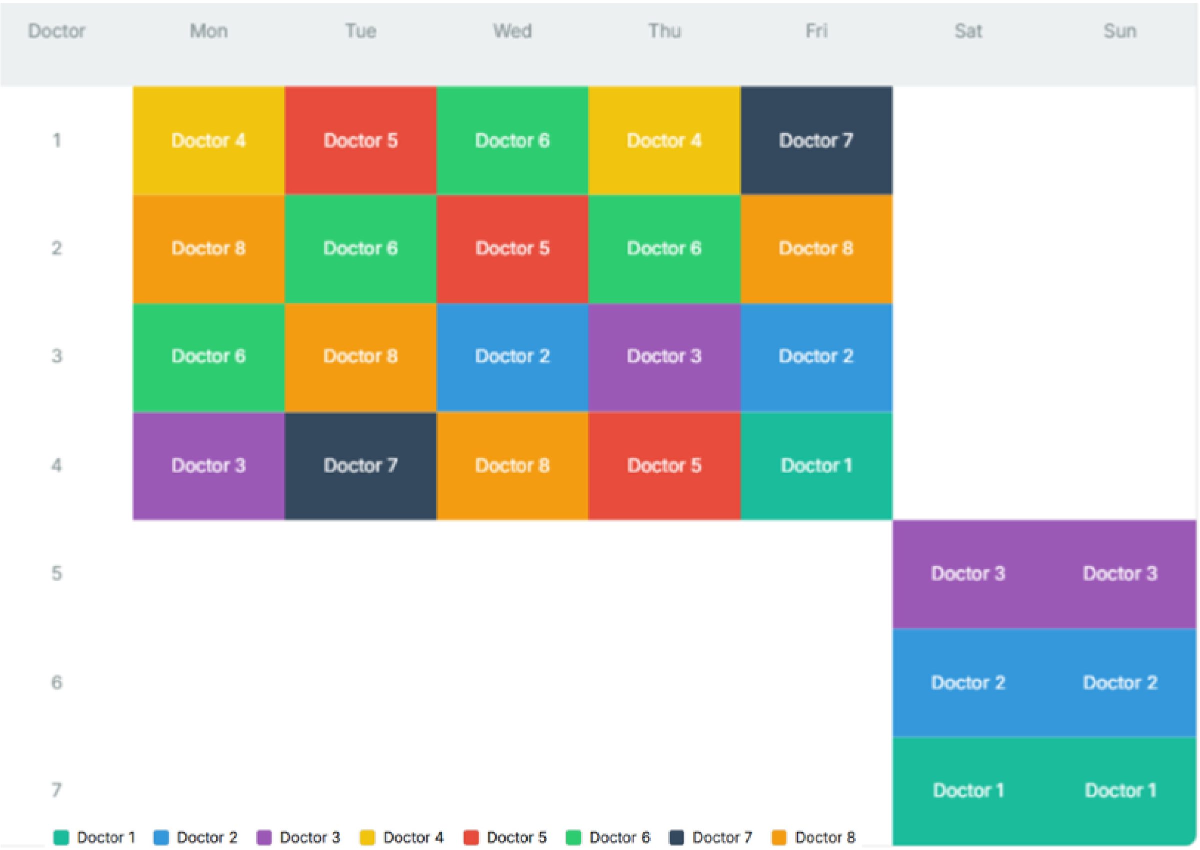

Figure 4.1: Work Schedule (without fairness) - Week 1

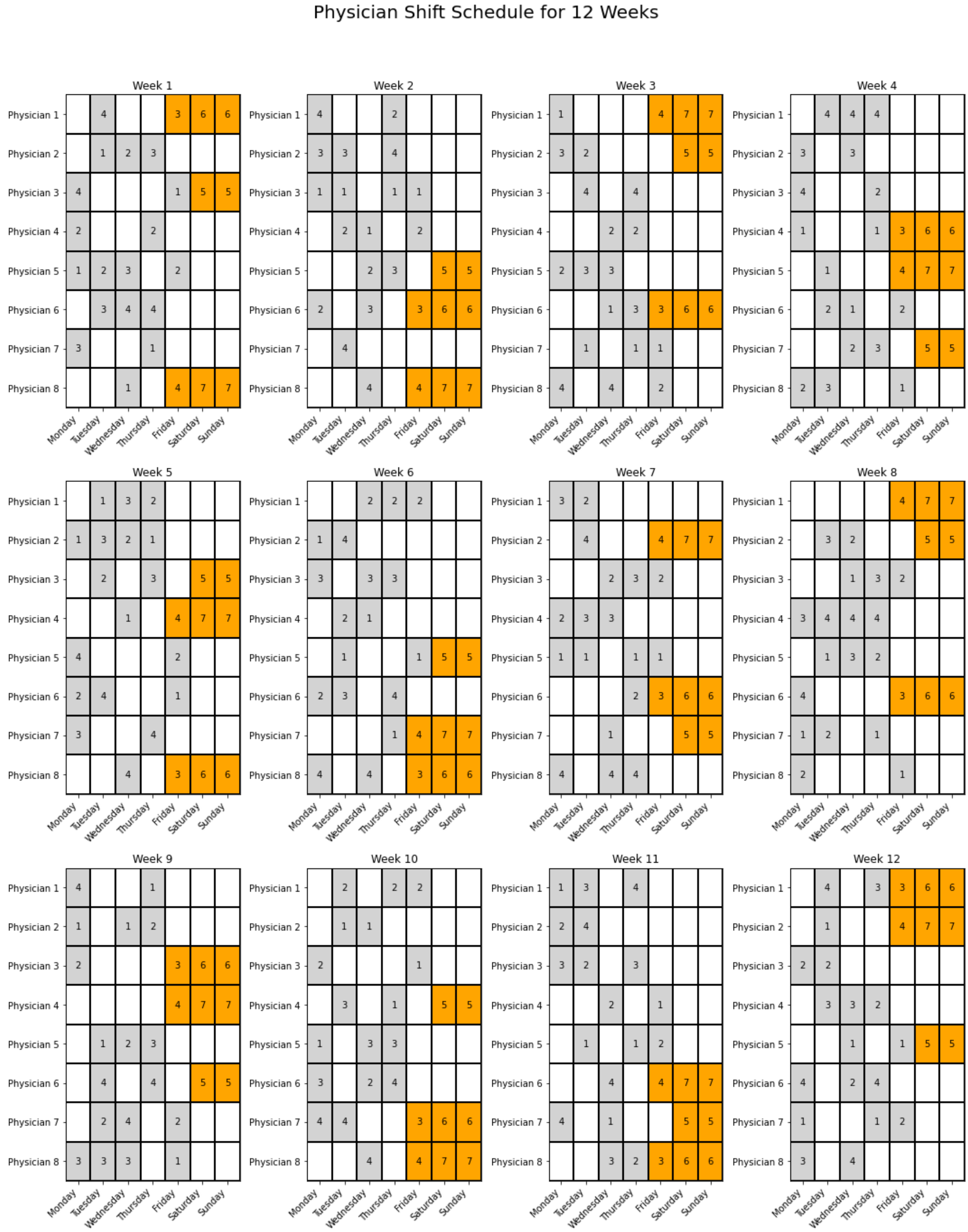

Figure 4.2: Work Schedule (without fairness) - twelve Weeks

Table 4.2 shows how the workload is split between the physician with the basic model, which means without the fairness. It can be clearly seen that the amount of assigned shifts varies between all possible pairings of the physicians and sums up to 98. The overall deviation in the weekend-work assignment is summing up to 34 and again reflects the deviations between each possible pairing of the physicians.

The emerging deviations of the solution for the first objective are disclosed in table 4.3. The biggest deviations are all located between physician three and six. The overall assigned shifts amount to eight additional shifts of work for physician six. Physician three at the same time is also the physician which is assigned to the least amount of weekend-work. Especially compared to physician six and eight, physician three is only assigned half the workload with three assigned weekends. Finally, the weekly hours diverge by 5.5 hours between the pairing of physicians three and six and thus this also forms the biggest deviance in this solution for this data point.

Overall, it is obvious that the fairness regarding the total workload in shifts and also weekly hours as well as the weekend-work distribution is lacking.

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	37	4	28.04
2	37	4	29.04
3	36	3	27.29
4	37	4	28.75
5	41	4	32.04
6	44	6	32.79
7	37	5	29.16
8	43	6	32.42

Table 4.2: Assigned workload for the first objective (without fairness)

Objective Value:	36
Total Shifts Deviation:	98
Total Weekend-Work Deviation:	34
Biggest Deviation Total Shifts:	8
Biggest Deviation Weekend-Work:	3
Biggest Deviation Weekly Hours:	5.5

Table 4.3: Deviations in the solution for the first objective (without fairness)

4.3 Solution for the second Objective

The extended model is able to find optimal results within 5.4 seconds. Further, it is able to find the fairest possible allocation without compromising the objective value for the optimal minimum weekend-work of 36, found in the first objective. Figure 4.3 represents an example for a schedule of the whole planning horizon solved for the second objective. The declaration on how to read this schedule is already depicted in section 4.2.

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	39	2	29.04
2	39	5	30.17
3	39	6	30.21
4	39	5	29.88
5	39	4	30.13
6	39	6	30.08
7	39	2	29.42
8	39	6	30.58

Table 4.4: Assigned workload for the second objective (with fairness)

Table 4.4 displays the workload assigned to each physician resulting from the extended model. Clearly, the assigned workloads are allocated evenly to all 8 physicians. Therefore, also no deviations in the shift allocations are emerging, as can be seen in table 4.5.

Objective Value:	36
Total Shifts Deviation:	0
Total Weekend-Work Deviation:	54
Biggest Deviation Total Shifts:	0
Biggest Deviation Weekend-Work:	4
Biggest Deviation Weekly Hours:	1.54

Table 4.5: Deviations in the solution of the second objective (with fairness)

The assigned weekend shifts are distributed very unevenly, especially between the physicians one and seven who only have to work on two weekends and physicians three, six and eight who, with six assigned weekends, have to work three times as often on weekends. Resulting from that, the biggest divergency in the weekend-work assignment is four weekends between the mentioned multiple physician pairings. Regarding the average weekly hours per physician there are only small deviations arising with 1.54 hours being the highest.

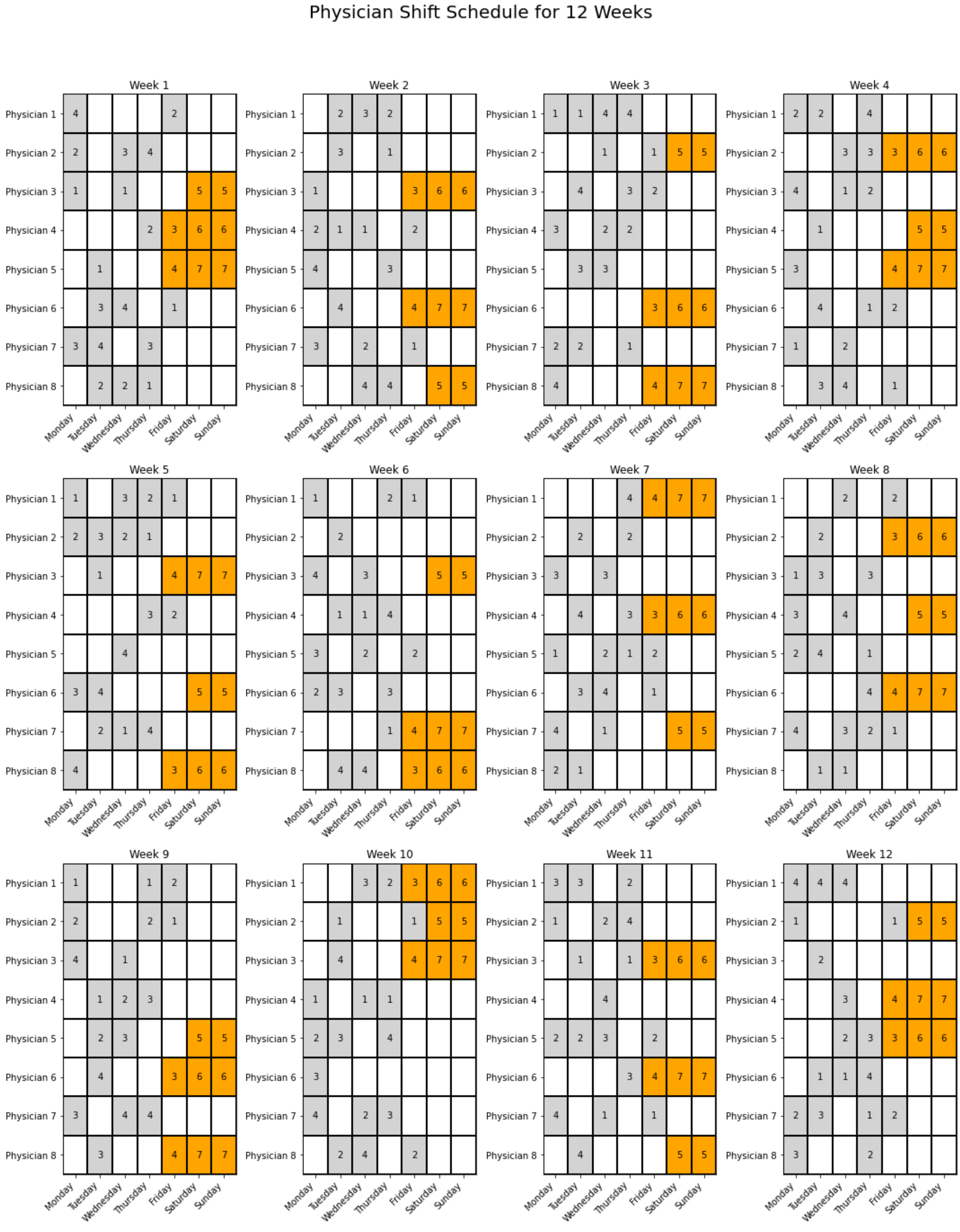

Figure 4.3: Work Schedule (with fairness) - twelve Weeks

4.4 Comparison of the Solutions

Reduction values discussed in this and the following sections always are meant as achievements from the use of the extended model over the basic model. When comparing both models, the time to execute is almost instant for both of them. The solution of the basic model does not consider a fair allocation of the workload whereas the solution of the extended model does. This shows very clearly when the assigned total shifts as well as the average working hours per week are compared in table 4.2 and table 4.4. Of course the showcased deviations in tables 4.3 and 4.5 are significant as well, as the total shift deviations are reduced from 196 to zero. The average weekly working time deviation is reduced by almost four full hours. Therefore, it can be concluded that the simple addition of the fairness constraint and alteration of the objective function definitely offer an easy and quick way to improve the fairness of a schedule significantly whilst retaining an almost identically quick runtime for this instance. The only point that has not improved is the uneven assignment of how often each physician has to work on a weekend within the planning horizon. This was to be expected though, as there is no constraint or objective which would approach this aspect.

Objective	Objective Value	Deviation Total Shifts	Deviation Weekend-Work	Gap (percent)	Runtime (seconds)
First	36	98	34	0.0	1.9
Second	36	0	54	0.0	5.4

Table 4.6: Comparison of the solution for the first and second objective

Objective	Biggest Deviation Total Shifts	Biggest Deviation Weekend-Work	Biggest Deviation Weekly Hours
First	8	3	5.5
Second	0	4	1.54

Table 4.7: Comparison of the biggest deviation for the first and second objective

Clearly this instance can be distributed very fairly as there are 26 shifts to be covered per week, multiplied by twelve weeks gives 312 shifts for the total planning horizon and that divided by eight results in an integer value of 39. Once the number of physicians considered for this instance changes to an uneven number like seven or nine, it will not be able to deliver a solution with a deviation of zero for total shifts. However, this is further discussed in the sensitivity analysis section below.

5 Sensitivity Analysis

In this section, a sensitivity analysis is performed to study the robustness of the model. All the cases covered in this analysis section use the data input from the real-world instance given in section 3 and also the same notations for the models are used. Then certain parameters are picked and changed one by one to see to which extent the model is still performing well and delivers solutions in a reasonable time span. The design of this analysis is built around and mainly focuses on seeing how a change in parameters can also impact the solvability and solution quality of the initial instance. Whether only the extended model or also the basic model have been used, depends on the specific parameter analyzed. This decision was also influenced by which findings and which comparisons seem relevant to be discussed in this section. A statement of what has been used is provided in each individual analysis section. Generally, it is a goal to show the impact that this fairness constraint has on different parameters and how much improvement can be gained by making use of it. Further, the impacts on the runtime of different parameter changes have been studied as well. To avoid difficulties, in the form of long runtimes, the runtime of the model was capped to one hour or 3600 seconds. Additionally, there is also the goal to learn which parameters of the given instance can be reduced or increased while still finding a feasible and, in the best case, also optimal solution for this specific physician scheduling problem. So in conclusion to look for buffers and bottlenecks that occur in this particular instance of a physician scheduling problem and discuss where a change of constraints or parameters could be of value. This is also contributing to study how versatile the model is for the same type of department in different hospitals.

The same comparison measures as described in subsection 4.1 and 4.4 have been used to create a fair environment for the sensitivity analysis of the models.

5.1 Workforce Reduction

For this part of the analysis the number of physicians has been reduced to seven, six and five. The solutions of the different workforce scopes then are summarized in a table and discussed in the corresponding comparison section below.

5.1.1 Seven Physicians

A workforce of seven physicians is sufficient to find a feasible solution. The corresponding allocations are displayed in table 5.1 and the deviations are showcased in table 5.2. This solution is not guaranteed to be optimal as the model stopped after the capped runtime. However, the solution is feasible and the optimal weekend-work of 36 was preserved, this means the gap of 25 per cent is fully ascribable to the total shift deviations. To offer a comparison of the improvement of the fairness aspect, this instance has also been used with the basic model and its results are showcased in table 5.3 and table 5.4. In this solution the proof of an optimal objective value of 36 for weekend-work is demonstrated again.

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	44	4	33.29
2	45	4	33.71
3	44	6	35.79
4	45	6	34.33
5	44	5	33.46
6	45	5	34.86
7	45	6	34.04

Table 5.1: Assigned workload for second objective (with fairness) - seven physicians

Weekend Work:	36
Total Shifts Deviation:	12
Total Weekend-Work Deviation:	22
Biggest Deviation Total Shifts:	1
Biggest Deviation Weekend-Work:	2
Biggest Deviation Weekly Hours:	2.5

Table 5.2: Deviations in the solution of the second objective (with fairness) - seven physicians

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	44	6	33.46
2	46	4	34.46
3	44	6	34.75
4	41	4	31.63
5	48	5	36.08
6	44	5	33.21
7	45	6	35.92

Table 5.3: Assigned workload for first objective (without fairness) - seven physicians

Weekend Work:	36
Total Shifts Deviation:	52
Total Weekend-Work Deviation:	22
Biggest Deviation Total Shifts:	7
Biggest Deviation Weekend-Work:	2
Biggest Deviation Weekly Hours:	4.45

Table 5.4: Deviations in the solution of the first objective (without fairness) - seven physicians

5.1.2 Six Physicians

The smallest possible physician reduction, without having to change any other parameters and still obtaining a feasible solution, is down to six. Here both the basic and extended model have been used each to see a difference in the fairness allocation. The deviation in total shifts between the two different models is very visible again. This can be seen in tables 5.5 and 5.7.

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	52	6	39.67
2	52	6	39.21
3	52	6	40.21
4	52	6	40.33
5	52	6	40.17
6	52	6	39.92

Table 5.5: Assigned workload for second objective (with fairness) - six physicians

Weekend Work:	36
Total Shifts Deviation:	0
Total Weekend-Work Deviation:	0
Biggest Deviation Total Shifts:	0
Biggest Deviation Weekend-Work:	0
Biggest Deviation Weekly Hours:	1.12

Table 5.6: Deviations in the solution for the second objective (with fairness) - six physicians

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	51	6	39.29
2	53	6	39.96
3	48	6	38.75
4	53	6	41.08
5	54	6	40.38
6	53	6	40.04

Table 5.7: Assigned workload for first objective (without fairness) - six physicians

Weekend Work:	36
Total Shifts Deviation:	36
Total Weekend-Work Deviation:	0
Biggest Deviation Total Shifts:	6
Biggest Deviation Weekend-Work:	0
Biggest Deviation Weekly Hours:	2.33

Table 5.8: Deviations in the solution for the first objective (without fairness) - six physicians

5.1.3 Five Physicians

The reduction to five physicians is not finding a feasible solution, where the rest of the parameters stay the same. The bottlenecks lie in the maximum weekly working time and the weekend-work allowed. Theoretically, with five physicians it would take at least 47.9 weekly hours per physician to satisfy the 239.5 hours of work, that need to be covered each week. This is obviously not possible with the given maximum weekly working time of 42.5 hours. Further to cover the weekend shifts it takes at least three physicians per week and twelve for four weeks. This is capped by two weekend shifts within four weeks per physician, which shows the need of a minimum of six physicians to cover this constraint and therefore the number of physicians in the workforce definitely creates a bottleneck, as expected.

5.1.4 Workforce Reduction - Comparison

When looking at tables 5.9 and 5.10 it can be seen that the objective value of the minimized weekend-work is never deviating from the original instance and therefore is staying at 36, which means that this objective value is not suffering at all from the reduction of physicians, even if the execution of the model is stopped by reaching the determined maximum runtime. Further, it is visible again that the extended model indeed is reducing the number of shift allocation deviations between the physicians every time. Another conclusion that can be drawn from these solutions is, that at least six physicians are needed to cover all shifts whilst also respecting all constraints. As this workforce scope is representing the bare minimum for this instance, it can be said, that the actual workforce of eight physicians is offering a good buffer to account for cases of illness or also a physician who takes one´s holiday.

Number of Physician	Objective Value	Deviation Total Shifts	Deviation Weekend-Work	Gap (percent)	Runtime (seconds)
7 (2nd Obj.)	36	12	22	25.0	3600.0
7 (1st Obj.)	36	52	22	0.0	6.3
6 (2nd Obj.)	36	0	0	0.0	88.0
6 (1st Obj.)	36	36	0	0.0	124.8
5	infeasible				

Table 5.9: Solution comparison of the different parameter inputs

Number of Physicians	Biggest Deviation Total Shifts	Biggest Deviation Weekend-Work	Biggest Deviation Weekly Hours
7 (2nd Obj.)	1	2	2.5
7 (1st Obj.)	7	2	4.45
6 (2nd Obj.)	0	0	1.12
6 (1st Obj.)	6	0	2.33
5	Infeasible		

Table 5.10: Biggest deviation comparison of the different parameter inputs

5.2 Planning Horizon - Extension and Reduction

For this part of the analysis the time horizon of the instance has been changed to four weeks, 52 weeks and 104 weeks to show how far it can be extended. Even a planning horizon of only a single week has been tried and an optimal solution was found. However, this was only for testing purposes and the results are not included for this analysis. The solutions of the tested planning horizons were summarized in tables and discussed in the corresponding comparison section below.

5.2.1 Four Weeks

The reduction of the planning horizon to four weeks is supposed to show the performance of the models for schedules that are planned for rather short periods and also how the fairness is affected by shorter periods. For this shorter time frame the improvement of the fairness in the allocations is especially noticeable when looking at tables 5.11 and 5.13. This improvement gets even more significant when looking at the biggest deviations in tables 5.12 and 5.14.

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	13	2	29.25
2	13	2	31.63
3	13	2	30.125
4	13	0	28.38
5	13	1	28.13
6	13	2	32.00
7	13	1	28.13
8	13	2	31.88

Table 5.11: Assigned workload for the second objective (with fairness) - four weeks

Weekend Work:	12
Total Shifts Deviation:	0
Total Weekend-Work Deviation:	22
Biggest Deviation Total Shifts:	0
Biggest Deviation Weekend-Work:	2
Biggest Deviation Weekly Hours:	3.88

Table 5.12: Deviations in the solution of the second objective (with fairness) - four weeks

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	5	1	12.88
2	16	2	38.63
3	17	2	38.63
4	17	2	39.25
5	17	2	36.00
6	11	1	25.38
7	12	1	27.38
8	9	1	21.25

Table 5.13: Assigned workload of the first objective (without fairness) - four weeks

Weekend Work:	12
Total Shifts Deviation:	146
Total Weekend-Work Deviation:	16
Biggest Deviation Total Shifts:	12
Biggest Deviation Weekend-Work:	1
Biggest Deviation Weekly Hours:	26.37

Table 5.14: Deviations in the solution for the first objective (without fairness) - four weeks

5.2.2 52 Weeks

The planning horizon of one year was picked to show how usable the model is for generating a schedule that is made for a long period in advance and how the fairness is affected in longer time periods. Table 5.15 shows the results achieved with the extended model and table 5.16 the associated deviations. A comparison of the assigned workloads and achieved deviations with the basic model can be made when looking at tables 5.17 and 5.18. Also for this time frame the deviations are visible.

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	169	19	29.63
2	169	17	29.50
3	169	18	30.12
4	169	20	29.77
5	169	22	30.10
6	169	19	30.50
7	169	20	29.97
8	169	21	29.92

Table 5.15: Assigned workload for the second objective (with fairness) - 52 weeks

Weekend Work:	156
Total Shifts Deviation:	0
Total Weekend-Work Deviation:	54
Biggest Deviation Total Shifts:	0
Biggest Deviation Weekend-Work:	5
Biggest Deviation Weekly Hours:	1.00

Table 5.16: Deviations in the solution of the second objective (with fairness) - 52 weeks

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	162	14	28.32
2	170	23	30.56
3	168	19	29.62
4	168	21	29.61
5	164	19	29.38
6	180	17	31.45
7	177	21	31.28
8	163	22	29.30

Table 5.17: Assigned workload for the first objective (without fairness) - 52 weeks

Weekend Work:	156
Total Shifts Deviation:	214
Total Weekend-Work Deviation:	96
Biggest Deviation Total Shifts:	18
Biggest Deviation Weekend-Work:	9
Biggest Deviation Weekly Hours:	3.13

Table 5.18: Deviations in the solution of the first objective (without fairness) - 52 weeks

5.2.3 104 Weeks

For this subsection the planning horizon was extended to create a schedule that is planning for two years in advance. This time horizon has been picked to showcase that the model has no issues with schedules planned for a very long time in advance and therefore it has been decided that the more complex extended model is sufficient to showcase the models capabilities and the comparison of both models would not contribute to showcase the solvability of such long planning horizons.

While planning for two years in advance actually seems rather unrealistic, the model is still capable of finding optimal schedules for such long time frames in a still very reasonable runtime. This can be seen in table 5.19 where only 294 seconds were needed to find an optimal schedule for the planning horizon of 104 weeks. The fairness of the shift allocation, as shown in table 5.20, is also still very well maintained.

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	338	35	30.27
2	338	46	30.43
3	338	38	30.06
4	338	36	30.00
5	338	37	29.77
6	338	37	29.20
7	338	40	29.59
8	338	43	30.19

Table 5.19: Assigned workload for the second objective (with fairness) - 104 weeks

Weekend Work:	312
Total Shifts Deviation:	0
Total Weekend-Work Deviation:	122
Biggest Deviation Total Shifts:	0
Biggest Deviation Weekend-Work:	11
Biggest Deviation Weekly Hours:	1.23

Table 5.20: Deviations in the solution of the second objective (with fairness) - 104 weeks

5.2.4 Planning Horizon - Conclusion

When looking at all tested time periods, it is very clear that the planning horizon of the schedule does not really play a crucial role when it comes to the performance and relative runtime of both models. Table 5.21 shows once again how much the consideration of fairness is able to even out the shift allocations between all physicians. Obviously, it is observable that a longer planning horizon leads to a bigger instance which needs a reflective longer runtime to create an optimal schedule. Further, it can be seen from the results of the basic model that the fairness of the allocations overall does also increase with the planning horizon. This can be seen when looking at table 5.22 where it is visible that the shortest planning horizon of four weeks performs worse with either model than the 52 week planning horizon regarding the biggest deviations. In detail this is reflected very well in the biggest deviation of the average working hours per week, as here the deviations of the basic model are even more drastic for shorter planning horizons with a biggest deviation of 26.37 hours for four weeks compared to only 3.13 hours for 52 weeks.

Planning Horizon	Objective Value	Deviation Total Shifts	Deviation Weekend-Work	Gap (percent)	Runtime (seconds)
4 (2nd Obj.)	12	0	22	0.0	0.7
4 (1st Obj.)	12	146	16	0.0	0.1
52 (2nd Obj.)	156	0	54	0.0	48.2
52 (1st Obj.)	156	214	96	0.0	36.3
104 (2nd Obj.)	312	0	122	0.0	294.0

Table 5.21: Solution comparison of the different parameter inputs - Planning Horizon

Planning Horizon	Biggest Deviation Total Shifts	Biggest Deviation Weekend-Work	Biggest Deviation Weekly Hours
4 (2nd Obj.)	0	2	3.88
4 (1st Obj.)	12	1	26.37
52 (2nd Obj.)	0	5	1.00
52 (1st Obj.)	18	9	3.13
104 (2nd Obj.)	0	11	1.23

Table 5.22: Biggest deviation comparison of the different parameter inputs - Planning Horizon

5.3 Maximum Consecutive Night Shifts - Reduction

To analyze whether the maximum allowed consecutive night shifts can lead to a bottleneck, the parameter for this constraint has been reduced and studied in this section. The reduction was supposed to be carried out until no feasible integer solution is found. As this reduction was able to be made down to the smallest possible parameter, one consecutive night shift, the analysis was made for three, two and as already mentioned one consecutive night shift. As the focus here is on the bottleneck rather than the impact on fairness created by the two models, only the extended model has been used to conduct this part of the analysis.

Three Consecutive Night Shifts

A reduction to one consecutive night shift less has not had any impact on the solution quality and lead to the same objective value of 36 for weekend work as well as the same deviation for total assigned shifts of zero which can be seen in table 5.24. The corresponding workload allocation can be seen in table 5.23.

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	39	3	28.88
2	39	5	29.83
3	39	5	31.54
4	39	6	30.50
5	39	4	30.79
6	39	6	28.88
7	39	2	28.63
8	39	5	30.46

Table 5.23: Assigned workload for the second objective (with fairness) - three night shifts

Weekend Work:	36
Total Shifts Deviation:	0
Total Weekend-Work Deviation:	46
Biggest Deviation Total Shifts:	0
Biggest Deviation Weekend-Work:	4
Biggest Deviation Weekly Hours:	2.16

Table 5.24: Deviations in the solution of the second objective (with fairness) - three night shifts

Two Consecutive Night Shifts

A parameter change to a maximum of two consecutive night shifts seems to create a problem with much higher complexity. The model only stopped after it reached the capped runtime of one hour and probably would have kept running further to confirm the optimality of the current solution in table 5.25 or to find the optimal solution. There is a need of more weekend-work, as the objective value for this has increased from 36 to 48, so by one third, compared to three or four allowed consecutive night shifts. All objective values for weekend-work and total shift deviations as well as the biggest deviations can be seen in table 5.26.

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	39	6	29.17
2	39	6	30.42
3	39	6	30.25
4	39	6	29.38
5	39	6	30.33
6	39	6	30.08
7	39	6	29.86
8	39	6	30.00

Table 5.25: Assigned workload for the second objective (with fairness) - two night shifts

Weekend Work:	48
Total Shifts Deviation:	0
Total Weekend-Work Deviation:	0
Biggest Deviation Total Shifts:	0
Biggest Deviation Weekend-Work:	0
Biggest Deviation Weekly Hours:	1.25

Table 5.26: Deviations in the solution of the second objective (with fairness) - two night shifts

One Consecutive Night Shift

Even a reduction to only one allowed consecutive night shift does still lead to a feasible, even optimal solution and therefore it can be said that the overall solvability of this instance is not affected by the allowed consecutive night shifts. Although the solution is guaranteed to be optimal and was found much faster with a runtime of only 5.4 seconds, it still has the same increased objective value for weekend work as the results for a maximum of two consecutive night shifts. The results of the allocation are shown in table 5.27 and the corresponding objective values and deviations in table 5.28.

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	39	6	30.08
2	39	6	30.46
3	39	6	30.38
4	39	6	29.92
5	39	6	30.13
6	39	6	29.58
7	39	6	28.88
8	39	6	30.08

Table 5.27: Assigned workload for the second objective (with fairness) - one night shift

Weekend Work:	48
Total Shifts Deviation:	0
Total Weekend-Work Deviation:	0
Biggest Deviation Total Shifts:	0
Biggest Deviation Weekend-Work:	0
Biggest Deviation Weekly Hours:	1.58

Table 5.28: Deviations in the solution of the second objective (with fairness) - one night shift

5.3.1 Maximum Consecutive Night Shifts - Conclusion

Overall, it can be said that this criterion of data input does not create a bottleneck for this instance in the physician scheduling problem when all other parameters remain unchanged. However, the quality of the solution is impacted by the reduction of maximum consecutive night shifts as shown in the comparison in table 5.29. Once the reduction is down to two, it does change the value of the minimum weekend work needed to satisfy this constraint and only allows for a minimum of 48 which was found within the capped runtime of one hour. The same objective value was then found to be optimal for only one consecutive night shift within only 5.4 second as can be seen in table 5.30.

Maximum Night Shifts	Weekend Work	Deviation Total Shifts	Deviation Weekend-Work	Gap (percent)	Runtime (seconds)
3	36	0	46	0.0	7.2
2	48	0	0	22.92	3600.0
1	48	0	0	0.0	5.4

Table 5.29: Solution comparison of the different parameter inputs - Night Shifts

Maximum Night Shifts	Biggest Deviation Total Shifts	Biggest Deviation Weekend-Work	Biggest Deviation Weekly Hours
3	0	4	3.88
2	0	0	2.37
1	0	0	1.58

Table 5.30: Biggest deviation comparison of the different parameter inputs - Night Shifts

5.4 Weekly Hours - Reduction

In this section, the maximum number of allowed working hours per week and physician have been reduced until no feasible solution could be found. For this instance a weekly workload of 32 hours per week and physician has proven to be infeasible and with 33 hours, the model is not capable of finding a single integer solution within the set capped runtime of one hour. Finally, a parameter of 34 hours was able to find an optimal solution within a very reasonable runtime of not even half a minute. As can be seen in all the solutions above, the average weekly workload with a workforce of eight physicians, is always located at around 30 hours. Therefore, 34 hours seem to be an accurate lower bound as not every week will be the same regarding work time for each physician due to the different constraints given. For the following analysis it was decided to use both models to see how much of an impact the aspect of fairness has when the workforce is reduced to the bare minimum for this instance.

34 Weekly Hours

The parameter change to a maximum weekly workload of 34 hours per physician was able to find an optimal solution within a very reasonable runtime of 26.2 seconds. Table 5.32 shows that the solution quality remained unchanged and therefore values of weekend-work and deviation in total shift assignments stayed the same. Deviations arising in the solution of the basic model can be seen in table 5.34. All allocations for the second model are displayed in table 5.31 and for the basic model in table 5.33.

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	39	5	29.04
2	39	5	29.29
3	39	6	31.33
4	39	4	30.21
5	39	3	30.00
6	39	5	30.63
7	39	5	30.25
8	39	3	28.75

Table 5.31: Assigned workload for the second objective (with fairness) - 34 Hours

Weekend Work:	36
Total Shifts Deviation:	0
Total Weekend-Work Deviation:	34
Biggest Deviation Total Shifts:	0
Biggest Deviation Weekend-Work:	3
Biggest Deviation Weekly Hours:	2.58

Table 5.32: Deviations in the solution of the second objective (with fairness) - 34 Hours

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	38	3	28.04
2	39	4	29.92
3	36	6	30.08
4	35	5	28.00
5	38	5	29.54
6	43	5	31.58
7	40	6	30.54
8	43	2	31.79

Table 5.33: Assigned workload for the first objective (without fairness) - 34 Hours

Weekend Work:	36
Total Shifts Deviation:	98
Total Weekend-Work Deviation:	46
Biggest Deviation Total Shifts:	8
Biggest Deviation Weekend-Work:	4
Biggest Deviation Weekly Hours:	3.79

Table 5.34: Deviations in the solution of the first objective (without fairness) - 34 Hours

5.4.1 Weekly Hours - Conclusion

As the weekly hours can be reduced to a maximum of 34 whilst still generating a feasible solution, it shows that there is a buffer of 8.5 hours per week and per physician to account for any unforeseen circumstances like a physician falling ill or other reasons causing staff shortages. Theoretically, there are 68 hours to cover for any unexpected events in every week, when in reality it has to be checked if the replacement allocation meets all the given constraints of the instance. Comparing the fairness when looking at tables 5.35 and 5.36, the extended model performed very well again and smoothed all shifts assignments despite the parameter change.

Weekly Hours	Weekend Work	Deviation Total Shifts	Deviation Weekend-Work	Gap (percent)	Runtime (seconds)
34 (2nd Obj.)	36	0	34	0.0	22.7
34 (1st Obj.)	36	98	46	0.0	11.9

Table 5.35: Solution comparison of the different parameter inputs - Weekly Hours

Weekly Hours	Biggest Deviation Total Shifts	Biggest Deviation Weekend-Work	Biggest Deviation Weekly Hours
34 (2nd Obj.)	0	3	2.58
34 (1st Obj.)	8	4	3.79

Table 5.36: Biggest deviation comparison of the different parameter inputs - Weekly Hours

5.5 Maximum Consecutive Days - Reduction

The parameter of maximum consecutive working days has been studied to find the lower bound with which the model is still solvable with an integer solution for this instance. The solvability of the model was granted to a reduction down to a maximum of four consecutive days. No feasible solution could be found for a maximum of three consecutive days.

Four Maximum Consecutive Days

The lower bound of solvability for this instance is located at a maximum of four consecutive working days. Optimality of this solution is not guaranteed though as the execution of the model was stopped due to reaching the time cap rather than proving optimality of the solution found. Further, the found solution shown in table 5.37 does have an increased value for weekend-work of 47, as can be seen in table 5.38. The allocation of the shifts is as fair as possible again with a total shift deviation of zero.

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	39	6	29.67
2	39	6	30.63
3	39	5	29.42
4	39	6	29.96
5	39	6	30.33
6	39	6	29.96
7	39	6	29.25
8	39	6	30.29

Table 5.37: Assigned workload for the second objective (with fairness) - four days

Weekend Work:	47
Total Shifts Deviation:	0
Total Weekend-Work Deviation:	7
Biggest Deviation Total Shifts:	0
Biggest Deviation Weekend-Work:	1
Biggest Deviation Weekly Hours:	1.38

Table 5.38: Deviations in the solution of the second objective (with fairness) - four days

Five Maximum Consecutive Days

When the parameter of maximum consecutive days is set to five, the model is capable of finding an optimal solution within only a few seconds, which is showcased in table 5.39. The exact runtime can be found in the corresponding conclusion part of this section. Additionally, the objective value of the weekend-work as well as total shift deviation was not affected by the reduction of this parameter as can be seen in table 5.40.

Physician	Assigned Total Shifts	Assigned Weekend-Work	Average Weekly Hours
1	39	3	29.75
2	39	5	31.42
3	39	6	30.00
4	39	5	27.75
5	39	5	30.71
6	39	3	29.29
7	39	4	30.75
8	39	5	29.83

Table 5.39: Assigned workload for the second objective (with fairness) - 5 Days

Weekend Work:	36
Total Shifts Deviation:	0
Total Weekend-Work Deviation:	34
Biggest Deviation Total Shifts:	0
Biggest Deviation Weekend-Work:	3
Biggest Deviation Weekly Hours:	6.67

Table 5.40: Deviations in the solution of the second objective (with fairness) - 5 Days

5.5.1 Maximum Consecutive Days - Conclusion

Concluding this section, it can be said that this parameter is indeed creating a bottleneck for this model and instance. Considering that this parameter can be halved from ten to five, without sacrificing any part of the solution quality, the parameter of ten in the initial instance can be seen as a buffer for unforeseen circumstances concerning workforce reduction. This is also reflected when looking at individual assignments in figure 4.2 or 4.3 where the working streak most of the time is around four days and is rarely longer. Further, when looking at the durations of the shortest shifts for weekdays and weekends, which are eight hours for weekdays and 8.5 hours for weekends, a maximum of five days can be assigned per week without exceeding the allowed weekly work time of 42.5 hours. This can be derived from dividing 42.5 by eight which allows for five assigned shifts and leaving only 2.5 hours per week, which are not sufficient for a sixth shift anyway. This means if five days at the end of one week and five at the beginning of the following week are assigned, the maximum weekly working hours are restricting the assignment to a maximum of ten consecutive days automatically, which makes the constraint for the maximum working streak redundant in the initial instance. To assure the two days off required after a ten day working streak is reached, the checking of how many days have been worked in a row is still necessary. A comparison of the solution quality of both tested parameters can be seen in table 5.41 and their biggest deviations in table 5.42.

Consecutive Days	Weekend Work	Deviation Total Shifts	Deviation Weekend-Work	Gap (percent)	Runtime (seconds)
4	47	0	7	23.4	3600.0
5	36	0	34	0.0	12.2

Table 5.41: Solution comparison of the different parameter inputs - Max. Cons. Days

Consecutive Days	Biggest Deviation Total Shifts	Biggest Deviation Weekend-Work	Biggest Deviation Weekly Hours
4	0	1	1.38
5	0	3	6.67

Table 5.42: Biggest deviation comparison of the different parameter inputs - Max. Cons. Days

6 Limitations

As this model basically delivers a solution under lab conditions, where everything is controlled for, it has to be acknowledged that there is data missing regarding unforeseen circumstances like cases of illness or sudden dismissals of physicians. Therefore, all of these results have to be considered under the circumstance of having a consistent workforce throughout the whole planning horizon. Unforeseen cases of illness of any physician are not accounted for but overall should be able be filled in by another physician due to the average buffer of 8.5 hours per physician per week, found in chapter 5 Sensitivity Analysis under the section 5.4 Weekly Hours - Reduction. Another consideration that has to be made is that in this model, any holiday each physician is entitled to, is not taken into account. However, again this should be solvable when the notice is given in time for the schedule planning and the assign variable of the concerned physician and days are set to zero through additional constraints.

7 Conclusion and Future Research

In conclusion, it can be said that this model offers a versatile way to find feasible and optimal solutions for the physician scheduling problem conquered in this paper. Even a very short runtime is achieved and therefore reaches the goal of reducing the workload and providing relief for the management and human resources of that department. The data input is variable and the model formulation does not need to be changed or altered as long as no additional constraints are needed. Further, the parameter changes are easy to conduct, as only the data input file needs to be updated with the new parameters.

After finding the extent of variability to which this model finds optimality in an efficient way, it can be said that it is definitely suitable for solving the physician scheduling problem for this instance. Therefore, also other hospitals or departments with a similar realm of parameters could benefit from this model.

Overall, the sensitivity analysis offered a good insight in how robust the model is for this instance and certain parameter changes. Major findings include that the reduction of the workforce is possible down to six, which means that two physicians can take their holiday at the same time and an optimal solution can still be found. When it comes to the analysis of the planning horizon, the model is reducible and extendable to any time frames needed. The allowed consecutive night shifts do not create an issue for the feasibility of the model when reduced, however the weekend work needed to satisfy this constraint does increase once only two night shifts allowed. As already discussed before, a minimum of three physicians is needed to cover the weekend shifts for this instance and if one of them covers all three night shifts on a weekend, it would require at least one more to cover all three night shifts when only two consecutively are allowed. This then of course is also altering how the other shifts belonging to the weekend can be assigned to the physicians whilst satisfying the ten hour break requirement. Ultimately the model stays feasibly and often optimally solvable. The allowed weekly hours can be reduced down to 34 for the used instance and it can still be solved to its optimum. This shows the buffer of 8.5 hours for the initial 42.5 hours of maximum weekly work time considered in the instance originally. The allowed maximum consecutive days also act as a buffer as this parameter can be reduced from ten to five without affecting the objective value for the weekend-work or total shift deviation.

Concluding for the sensitivity analysis part, bottlenecks were found for the number of physicians in the workforce and the weekly working hours per physician. These resulting limiting factors were to be expected though, as these are the two criteria that enable the shifts to be covered in the first place. E.g. if the workforce scope is to low, a schedule would never be feasible as the allowed weekly working time will be exceeded at some point. Another bottleneck was located in the maximum allowed working streak, as this of course then is also limiting the hours that are able to be worked per week depending on the days off that have to be given after a streak is reached.

The extended model definitely does introduce much more balanced work assignments no matter the parameter changes and should be applied over the basic model. This conclusion is made as the working assignments should be as fair as possible to enable a positive and good working environment and therefore retain the workforce as long as possible. As already mentioned in the introduction section of this paper, this is a substantial goal for hospitals as it is rather hard to find new physicians to replenish the workforce. When an instance reaches the capped runtime and the optimality of the so far found solution is not guaranteed, it can be beneficial to use the basic model as well to see the optimality for the value of weekend-work. This solution then can be used to compare how far this value is optimized in the extended model and if the gap of the solution quality is caused by the total shift deviation or not. Ultimately there is value in using both models when optimality is not guaranteed in a solution of the extended model.

Genuinely improvements in the fairness in the allocations of overall shifts and therefore overall average working hours can be made with the use of the extended model. However, it does not enable a fair assignment of how many weekends are assigned to each physician. The total minimization of weekend work is given but the assigned amounts per physician still often vary a lot between all physicians. This depicts a potential problem that could be looked at in future problems, where constraints and objectives are formulated to also balance the allocation of how many weekends each physician has to work. This would additionally improve the working environment as everyone is treated as equal as possible in this regard as well.

List of Figures

List of Tables

Bibliography

Addou, I. and Soumis, F. (2007). Bechtold-Jacobs generalized model for shift scheduling with extraordinary overlap. *Annals of Operations Research*, 155(1):177–205.

Adhiana, T. P., Krisnawati, M., Larasati, A. D., and Sofiana, A. (2023). Integer programming method for employee shift scheduling problem. page 090014, Perlis, Malaysia.

Aiken, L. H. (2002). Hospital Nurse Staffing and Patient Mortality, Nurse Burnout, and Job Dissatisfaction. *JAMA*, 288(16):1987.

Aykin, T. (2000). A comparative evaluation of modeling approaches to the labor shift scheduling problem. *European Journal of Operational Research*, 125(2):381–397.

Beaulieu, H., Ferland, J. A., Gendron, B., and Michelon, P. (2000). A mathematical programming approach for scheduling physicians in the emergency room. *Health Care Management Science*, 3(3):193–200.

Bruni, R. and Detti, P. (2014). A flexible discrete optimization approach to the physician scheduling problem. *Operations Research for Health Care*, 3(4):191–199.

Carter, M. W. and Lapierre, S. D. (2001). Scheduling Emergency Room Physicians. *Health Care Management Science*, 4(4):347–360.

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*, 103:107151.

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):1–18.

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

Gendreau, M., Ferland, J., Gendron, B., Hail, N., Jaumard, B., Lapierre, S., Pesant, G., and Soriano, P. (2007). Physician Scheduling in Emergency Rooms. In Burke, E. K. and Rudová, H., editors, *Practice and Theory of Automated Timetabling VI*, volume 3867, pages 53–66. Springer Berlin Heidelberg, Berlin, Heidelberg. Series Title: Lecture Notes in Computer Science.

Geneva: World Health Organization (2023). Workload indicators of staffing need: user's manual, second edition.

Gunawan, A. and Lau, H. C. (2013). Master physician scheduling problem. *Journal of the Operational Research Society*, 64(3):410–425.

Jaumard, B., Semet, F., and Vovor, T. (1998). A generalized linear programming model for nurse scheduling. *European Journal of Operational Research*, 107(1):1–18.

Kletzander, L. and Musliu, N. (2020). Solving the general employee scheduling problem. *Computers & Operations Research*, 113:104794.

Legrain, A., Bouarab, H., and Lahrichi, N. (2015). The Nurse Scheduling Problem in Real-Life. *Journal of Medical Systems*, 39(1):160.

Mansini, R. and Zanotti, R. (2020). Optimizing the physician scheduling problem in a large hospital ward. *Journal of Scheduling*, 23(3):337–361.

Nahmias, S. (2009). *Production and operations analysis*. The McGraw-Hill/Irwin series operations and decision sciences. McGraw-Hill Irwin, Boston, 6th edition edition.

Ogulata, S. N., Koyuncu, M., and Karakas, E. (2008). Personnel and Patient Scheduling in the High Demanded Hospital Services: A Case Study in the Physiotherapy Service. *Journal of Medical Systems*, 32(3):221–228.

Santos, H. G., Toffolo, T. A. M., Gomes, R. A. M., and Ribas, S. (2016). Integer programming techniques for the nurse rostering problem. *Annals of Operations Research*, 239(1):225–251.

Trilling, L., Guinet, A., and Magny, D. L. (2006). NURSE SCHEDULING USING INTEGER LINEAR PROGRAMMING AND CONSTRAINT PROGRAMMING. *IFAC Proceedings Volumes*, 39(3):671–676.

Turhan, A. M. and Bilgen, B. (2020). A hybrid fix-and-optimize and simulated annealing approaches for nurse rostering problem. *Computers & Industrial Engineering*, 145:106531.

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

Villarreal, M. C. and Keskinocak, P. (2016). Staff planning for operating rooms with different surgical services lines. *Health Care Management Science*, 19(2):144–169.