Norwegian School of Economics

Bergen, Fall, 2022

Optimal Scheduling of Nursing Shifts

A Case Study on Work Scheduling at Haukeland University Hospital

Natalie Alsaker Bergsvik & Linnéa Marie Wahlman Bjøru

Supervisor: Mario Guajardo

Master Thesis: Economics and Business Administration

Major: Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

Acknowledgements

This thesis is written as part of our Master of Science in Economics and Business Administration, with a specialization in Business Analytics, at the Norwegian School of Economics (NHH). The process of creating this thesis has been challenging while simultaneously highly educational and profoundly rewarding.

We would like to start by expressing our sincere gratitude to our supervisor, Mario Guajardo, who has encouraged and inspired us throughout the process. He has been an excellent guide and motivator, with a contagious curiosity that inspired us to keep improving, asking new questions and truly make the most of this semester.

We would also like to thank the staff at Haukeland University Hospital for their tremendous help in making this thesis possible. A special thank you goes out to Wibeke Krokås, advisor at Haukeland, for providing the case that this thesis is based on. Wibeke has continually helped us throughout the semester, providing crucial information and answering our many questions. She has given us a deeper understanding regarding nurse scheduling at the hospital, and this thesis would not have been possible without her.

Lastly, we would like to thank our friends and family who have been there for us throughout the semester. Especially, we would like to thank our friends who work in nursing, for their advice and opinions that helped create a schedule which strives for a high level of staff satisfaction.

Thank you.

Abstract

This thesis presents a nurse scheduling problem tailored to characteristics common of Norwegian hospitals. The problem involves allocating nurses to specific shifts to ensure coverage of demand, while respecting work regulations and accounting for balance of workload, general nurse preferences and fairness. We formulate models for the nurse scheduling problem in line with the scheduling principles at Haukeland University Hospital and solve them using mathematical programming techniques. The purpose lies in the attempt to present a more efficient approach to the problem, compared to the manual scheduling approach currently utilized at the hospital.

The method involves formulating a mixed integer programming model, which is implemented computationally in an optimization software. Multiple decision models are produced to represent two different forms of schedules, one cyclical and one calendar based. The model for the cyclical schedule can be optimized directly using the solver of the software. The model for the calendar-based schedule is more complex and is therefore solved by designing a decomposition heuristic approach to find a good solution in a reasonable computational time.

The conclusion is that the schedules derived from the decision models are viable, with emphasis on the considerable time savings compared to the current scheduling approach at Haukeland. Currently, the hospital uses a manual method which takes approximately four to six weeks to create a schedule, whereas the models proposed in this thesis are able to derive an optimal solution within two hours. The models manage to effectively account for many criteria, including work regulations, fairness, balance of workload and preferred practices. The work in this thesis has been conducted through close cooperation with a representative from the staffing department at Haukeland, and the solutions derived from the models are able to capture their considerations in practice to a large extent. Our work has contributed to giving staff at Haukeland insight on how optimization and computational tools can be used to deal with their complex work scheduling problem.

Contents

List of Figures

List of Tables

1. Introduction

One of the fundamental challenges in a hospital is efficient and effective scheduling of nursing personnel. The nurses play a vital role in the hospital by providing care, support, and treatment for the sick and supporting their families. Thus, the scheduling of nursing personnel is a critical component, as the final schedule should result in timely and high-quality patient care. Furthermore, unattractive schedules have been associated with turnover intention among nurses (Blytt et al., 2022). Meanwhile, there is already a shortage of healthcare workers in Norway (Dolonen & Reppen, 2022). The insufficiency is especially critical when it comes to nurses, accounting for more than one third of the shortage according to NAV, the Norwegian Labour and Welfare Administration (Myklathun, 2022, p.8). To add to the challenge, the increasing population longevity will lead to rising demand for healthcare services in the future. Consequently, the recruitment and retention of medical expertise is crucial for the population's healthcare needs to be met. Moreover, Ose et al. (2022) suggests that high-quality schedules can reduce work-related sick leave among nurses.

A solid nurse schedule is clearly a crucial component, as it can be linked to quality of patient care as well as turnover intention and sick leave among nurses. However, composing a schedule is a highly complex task due to a vast range of criteria that must be considered. To maintain safe operations, the schedule must ensure that demand is covered, often around the clock, while still respecting work regulations. Furthermore, a good schedule would take into consideration the well-being and contentment of the nurses. This may include accounting for balance of workload, fairness, preferences and facilitating for personal inclinations due to health or other circumstances. Traditionally, nurse schedules have been manually planned and constructed with limited tools and resources available. This is a highly time-consuming process considering the complexity of the problem and finding the best solution with respect to all criteria is nearly impossible to do manually. The need for a more efficient approach is evident, as this could free up time and allow for better allocations of human and financial resources.

The purpose of this master's thesis is to solve a nurse scheduling problem by formulating a mathematical optimization model that can create realistic and satisfactory schedules in a way that is preferable to manual methods.

In cooperation with the staffing department at Haukeland University Hospital, we were provided with a realistic case replicating a typical department at Haukeland. The common practice at Haukeland today is to create schedules using manual methods. In this thesis, we utilize optimization and mathematical programming to solve the nurse scheduling problem provided to us. Automated approaches, such as optimization, hold significant potential for decreasing the time required for the scheduling process (Burke et al. 2004). In addition, such approaches can produce several viable solutions while accounting for qualitative factors such as fairness and balance of workload. Haukeland University Hospital has a strong culture of considering employee preferences and creating schedules which aim to promote the well-being of their employees. We intend to reflect these values in the optimization models, resulting in a highly detailed scheduling problem.

This thesis is divided into 9 chapters. In this chapter we have clarified the purpose of the thesis, as well as the motivation behind the chosen topic. Chapter 2 aims to give the reader an understanding of the elements in mind when creating nurse schedules at Haukeland University Hospital. In chapter 3 we discuss relevant literature related to nurse scheduling and our chosen method. This is followed by chapter 4, which provides a detailed description of the nurse scheduling problem. In chapter 5 we present the mathematical optimization models formulated for the nurse scheduling problem. The data used in the models is presented in chapter 6, along with the implementation of the models in a mathematical software. In Chapter 7 we present and discuss the results obtained. Chapter 8 discusses limitations of the models and presents suggestions for future research. In chapter 9 conclusions are drawn based on the discussion of results in chapter 7.

2. The Scheduling Process at Haukeland

This chapter will explain the steps involved in the planning and scheduling process at Haukeland. The chapter is based on the information we have received from the staffing department at the hospital.

Haukeland mainly utilizes two forms of schedules, cyclical or calendar-based, depending on the needs of the department. A cyclical schedule is characterized by having a planning horizon with a certain number of weeks, and once those weeks have passed, the schedule is repeated. Consequently, each nurse works a cyclical shift pattern. A calendar-based schedule is not repeated, and usually has a longer planning horizon than a cyclical schedule. This allows for more flexibility by accounting for variations in workforce requirements, and planning vacations, holidays, or other considerations that is associated with particular dates throughout the period.

The average time to manually compose a cyclical schedule is estimated to take approximately 1-2 weeks, however, this can vary depending on the complexity of the department structure. The process is more complex and time consuming when creating a calendar-based plan as opposed to a cyclical schedule with a shorter planning horizon. This is due to the longer planning horizon, which allows for planning e.g., vacations, holidays, and meetings. Furthermore, the employees can make requests for certain shift patters, off days and vacations. It is less common to account for individual wishes or requests from the employees in a cyclical schedule. A cyclical schedule needs so called “assisting schedules” to account for special times of the year such as Christmas and summer. In this thesis we will not consider or create any assisting schedules. The figure below illustrates the planning process at Haukeland regarding a 52-week calendar-based schedule.

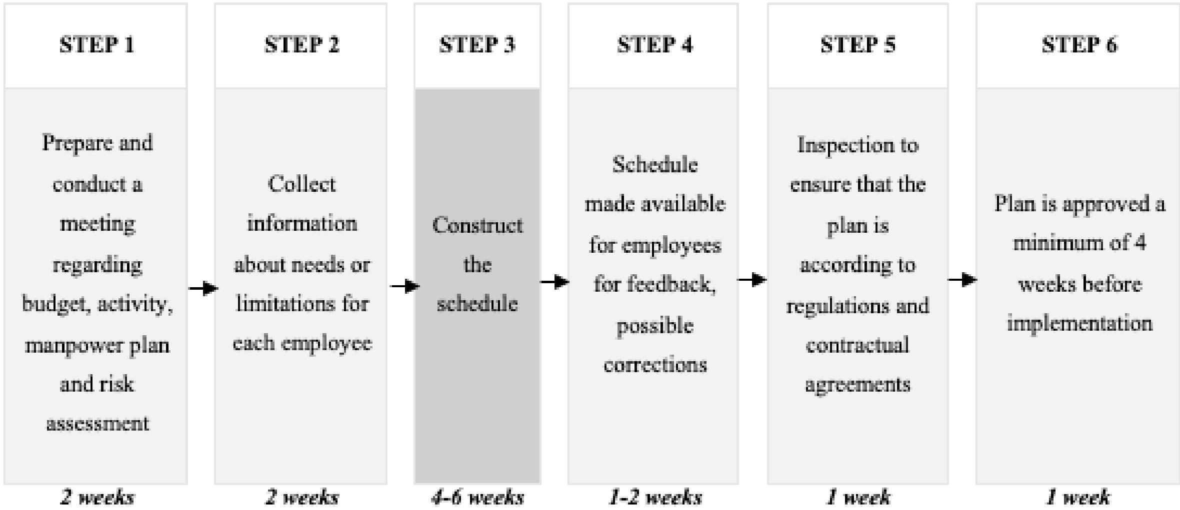

Figure 2.1*: The scheduling process at Haukeland University Hospital*

Step 1 involves preparing and conducting a meeting based on the budget, activity, manpower plan and risk assessment. The ward manager, union representatives and safety representatives are involved in this step. The participants collect sufficient information about conditions at the unit that are important in relation to the schedule. The purpose of this step is to review and agree on considerations that must be accounted for when preparing the schedule, e.g., work regulations and preferred practices. Furthermore, clarification of upcoming steps in the planning process is outlined. This process takes about two weeks to conclude.

Information regarding employee limitations is collected in step 2. Some employees might have certain limitations because of their health or other circumstances. Additionally, individual request and wishes are collected and evaluated. The degree to which personal requests are granted depends on the department and implementation feasibility. Safe operation must always be the priority to ensure sufficient patient care. Another two weeks is reserved for this step.

The construction of the schedule becomes the matter in step 3, after all necessary information and considerations are gathered and determined. The method involves manually assigning nurses to shifts to ensure demand is covered, while respecting all considerations determined in step 1 and 2. A scheduling software is utilized as a guidance, which provides important information and notifies the user if any hard constraints, such as work regulations, are violated. However, the actual assignment of shifts is done manually, which Haukeland refers to as plotting. This is a highly time consuming and demanding task, estimated to take about four to six weeks to finish. In an attempt to simplify the job, a certain strategy is commonly followed. This involves first assigning vacations, work weekends and holidays. Subsequently, night shifts are assigned, followed by evening shifts, and lastly, day shifts.

After completing the schedule, it is made available for the employees to review and provide feedback in step 4. Any adjustment based on the feedback may be implemented where applicable. This step should be finished within 1-2 weeks. In step 5, the ward manager and union representatives revise the schedule as a final inspection to ensure regulations and contractual agreements are enforced. In the last step, the finished schedule should be approved within 4 weeks of implementation.

In this thesis, we will create both a cyclical schedule and a calendar-based schedule using optimization techniques. The purpose of the optimization models is to replace the most comprehensive stage, the construction of the schedule in step 3. All steps before and after would still be crucial, even with the help of an automated method to compose the schedule. Step 1 and 2 would be necessary to know what to implement in the algorithm. Further, a final schedule should always be quality checked in step 4 before approval in step 5. Nevertheless, an approach that could speed up the process in step 3 would make a considerable difference in terms of time saving and resource allocation. Furthermore, the potential of an optimization model regarding fairness and nurse satisfaction could be exploited to the maximum, as the algorithm may find the optimal solution a manual method would merely be lucky to discover, but most likely never would.

3. Literature Review

This chapter will provide an overview of relevant terms commonly used in the literature regarding personnel scheduling. Further, we discuss previous research papers related to the topic of scheduling using mathematical programming.

In the last few decades, a considerable amount of research has been performed on the topic of personnel and nurse scheduling. The extensive interest in the problem could be explained by its complexity, challenging features, and practical relevance (Brucker et al., 2011). Scheduling of healthcare personnel, e.g., nurses, is particularly challenging. Consequently, different approaches have been explored to deal with this problem. In the literature, automated methods are commonly categorized as exact or heuristic methods (Burke et al., 2004; Ernst et al., 2004; Bergh et al., 2013). Exact methods are characterized by guaranteeing the optimal solution to a particular problem given certain goals and constraints. Mathematical programming, also referred to as optimization, is a well-known exact method. A Mathematical optimization model consists of decision variables, an objective function to be minimized or maximized, and a set of constraints that must be satisfied. Based on the model formulation, an optimal solution is found. However, for very complex problems, exact methods may require a very long computational time to find the optimal solution. In those situations, heuristic approaches could be more beneficial. Heuristic approaches are designed to find good solutions within a reasonable time frame but cannot guarantee that the solution is the global optimum. Considering the complexity of many nurse scheduling problems, heuristics appear to be favorable to exact methods in the literature (Burke et al. 2004).

It is common to distinguish between hard and soft constraints. Soft constraints can be violated, but its violations are associated with a form of penalty. Constraints regarding preferred practices and nurse satisfaction, for example, are often addressed using soft constraints (Bergh et al. 2013). On the contrary, hard constraints e.g., work regulations, are non-negotiable and must always be satisfied. The main distinction between these two categories is that a hard constraint can make a problem infeasible, but a soft constraint will not. Whether a constraint is formulated as hard or soft depends on the problem that is being solved and its specific goals and characteristics.

The origin of the personnel scheduling problem can be traced back to the work of Edie (1954), which was further investigated by Dantzig (1954) using mathematical programming (Akkermans et al., 2021). Although the personnel scheduling problem has evolved considerably since then, Danzig's contribution to the literature is still very attractive to researchers (Bergh et al., 2013). It has been applied to numerous areas, including health care, and laid the foundation for further research regarding personnel scheduling. However, Dantzigs approach is best suited to formulate simplified scheduling problems, and it is difficult to accommodate for more complicated characteristics that often occur in real-life personnel scheduling problems (Akkermans et al., 2021).

Stølevik et al. (2011) developed a mathematical model formulated to meet performance requirements and functional specifications typical for Norwegian hospitals. The objective is to minimize the difference between contracted and scheduled hours, as well as minimizing violations of the soft constraints. The difference in contracted and scheduled hours are squared, such that many small deviations are preferred to fewer and larger. Furthermore, Stølevik et al. (2011) include a hard constraint limiting scheduled hours to be within a range of +/- a certain percentage, *p*, to the contracted hours. We found one of their constraint formulations to be particularly helpful. This constraint concerned mandatory rest time between two shifts, which meant only a certain pair of shifts were allowed on consecutive days. We implemented their formulation of this constraint, with minor adjustments to adapt it to our problem. The work of Stølevik et al. (2011) has been a valuable resource and, to the best of our knowledge, it's one of the first papers that tailors the nurse scheduling problem to Norwegian hospitals using mathematical programming. We draw inspiration from their work but widen the problem scope by adapting it to a specific case and adding new components such as holidays and vacations.

Another paper that concentrates the nurse scheduling problem to a Norwegian hospital is conducted by Beckmann & Klyve (2016) as their master thesis. They develop an integer programming model to solve a real nurse scheduling problem for the Maternity Ward West department at St. Olavs hospital in Trondheim, Norway. Their problem concerns the scheduling of 69 employees over a planning horizon of 27 weeks. This planning horizon is considerably longer than most other papers throughout the literature (Burke et al. 2004). Like our calendar-based model, Beckmann & Klyve (2016) also includes holidays and vacations. Their objective function maximizes the number of respected employee requests and the number of desirable shift patterns allocated. They manage to capture the importance of nurse satisfaction by emphasizing the individual preferences in their model. The difference in contracted hours and scheduled hours is not a part of the objective function. Instead, it is addressed using a hard constraint limiting the scheduled hours to stay within a certain range to the contracted hours, similar to the hard constraint in Stølevik et al. (2011). Beckmann & Klyve (2016) inspired us regarding how to tackle the important "F1" constraint, which makes sure that each nurse has a minimum continuous free period every week. Some adjustments were necessary to adapt it to our problem and the program we are using, but it certainly influenced how we eventually formulated this constraint.

Zanda et al. (2018) solved a nurse scheduling problem with a yearlong planning horizon using a decision support system based on linear integer programming. The key aspects in the paper are motivated by a real case study, namely the surgery department of the University Hospital in Cagliari, Italy. They are minimizing the number of nurses in deficit, minimizing the difference in expected work hours and scheduled work hours, and prioritizing a schedule where off days are assigned consecutively. Their problem scope shares some fundamental similarities to the calendar-based plan in this thesis. Many characteristics are comparable, notably the planning horizon, as well as accounting for vacations and holidays. They approach the problem using an optimization-based heuristic and create a calendar-based schedule for a whole year. They decompose the problem by breaking the main problem of one year into smaller subproblems relative to one month. All sub-problems consider decisions made in the previous period, while receiving information to be implemented in the current period. Initially, we considered approaching our problem the same way. However, we wanted to explore other alternatives.

In this thesis we resort to both an exact and a heuristic approach depending on the problem. The cyclical schedule is solved with an exact approach using mathematical programming. An optimal solution is obtained within a reasonable time frame. We address fairness, preferred practices and personal inclinations using a combination of hard and soft constraints, where violation of the soft constraints is penalized in the objective function. For the calendar-based schedule we utilize an optimization based heuristic approach by decomposing the problem into two sub problems. The first model assigns vacations and work weekends, and if the nurses work night in that weekend or not. The second model allocates shifts using the result from stage-1 as fixed inputs, and ultimately produces the final schedule. The goal in the first model is to find a feasible solution while respecting a series of hard constraints. The goal in the second model is to minimize undesirable shift patterns and unbalanced workload.

As already established, the nurse scheduling problem has been extensively researched. However, Burke et al. (2004) points out that most of them involves a relatively short planning horizon of a few weeks, typically four. As the planning horizon expands, the computational effort increases. This became clear when we attempted to expand the cyclical model to encompass a planning horizon of one year. Based on our own research, papers that specifically explore approaches to reduce the computational complexity caused by longer planning horizons, are still outnumbered. This could be because cyclical schedules, or calendar-based schedules with a shorter planning horizon, appear to be most common in the health care industry from a global perspective. However, calendar-based schedules with a planning horizon of 52 weeks seem to be of increasing interest in Norway (Berland, 2019). Hence, we recognized the need for an approach which addresses this problem. The novelty of this thesis lies mainly in the attempt to appropriately address the calendar-based nurse scheduling problem with a longer planning horizon.

4. Problem Description

Nurse scheduling problems can differ a lot in terms of scope and scheduling rules. In this chapter, we describe the nurse scheduling problem relevant in this thesis. The scheduling rules are determined based on general scheduling principles at Haukeland University Hospital. We solve the nurse scheduling problem for two forms of schedules, a cyclical schedule and a calendar-based schedule. Section 4.1 describes the general scheduling rules applicable to both schedules. Section 4.2 presents rules regarding the cyclical schedule, and the rules regarding the calendar-based schedule is addressed in section 4.3. Lastly, section 4.4 presents a compressed summary of all scheduling rules.

4.1 General Scheduling Rules

This section describes the general scheduling rules. These rules will apply to both the cyclical and calendar-based schedule unless stated otherwise in each problems designated subchapters.

4.1.1 Employees and Work Hours

A department typically has a mix of both full-time and part-time nurses, and every nurse has a specific number of average work hours stated in their contract. For a full-time nurse, this is equivalent to an average of 35.5 hours per week. Ideally, the scheduled hours should be as close to the contracted hours as possible. A slight deviation between contracted and scheduled hours is allowed. However, scheduled hours should not be allowed to exceed contracted hours, as this would imply planning for nurses to work overtime. Further, some of the nurses are limited as to when and what shifts they are allowed to be assigned to. For example, a group of nurses only work during the weekends, which we will refer to as weekend nurses, and some nurses only work night shifts, which we will refer to as night shift nurses.

4.1.2 Shifts and Demand

The schedule consists of work shifts and off shifts, and each nurse must be assigned one shift daily. The different kinds of off shifts will vary depending on the problem, but an “F”-shift and an F1-shift is included in both. These will be described in more detail in the section about rest time.

Work shifts can take place during the day, evening, or night. This case is limited to one day, one evening, and one night shift, which are often abbreviated to D-shift, E-shift, and N-shift respectively. Each work shift must be covered by a specific number of nurses for each day of the week. This is commonly called the manpower plan, staffing plan or demand.

Some days and periods during the year will deviate from the manpower plan. This is not accounted for in the cyclical schedule but will be considered in the calendar-based schedule. Such days are typically holidays and specific periods during the summer. We address this further in section 4.3. Otherwise, the total number of nurses assigned to a specific shift must cover the demand. One additional nurse is allowed during day shifts only. This might be necessary to achieve the right amount of work hours, but if so, the nurses usually prefer to be assigned to a day shift rather than an evening or night shift.

4.1.3 Weekends

The scheduling rules regarding weekends are different in the cyclical and yearly schedule and will be addressed more detailed in each problem´s designated chapter. However, some restrictions are constant in both problems. All shifts starting on Saturday and Sunday, as well as the evening shift on Friday, is considered part of the weekend. Consequently, E-shifts on Fridays can only be allocated to nurses working the following weekend. If a nurse works a weekend, they must work both Saturday and Sunday, and can be assigned the evening shift on Friday. This means a nurse cannot work merely one day during the weekend. Furthermore, nurses work either night shift starting on Friday, Saturday and Sunday, or the nurse does not work night shifts during the weekend at all.

The weekend nurses are oftentimes students, so to always ensure enough experience among the staff a maximum number of weekend nurses can be assigned the same weekend, but not the same shift. The frequencies of weekends are managed differently depending on the problem and will be addressed further in section 4.2 and 4.3.

4.1.4 Regulations and Policies

The schedules must uphold several rules and regulations set by the government. These rules exist to take the employees' health and well-being into consideration. Furthermore, most facilities have their own preferable practices, Haukeland being one of them, which we aim to implement. These rules and preferable practices generally regard rest time and balance of workload.

Rest Time

For each week, a continuous off period of at least 35 hours must be planned, which extends over a full day. In Norway, this is commonly referred to as the F1 day. Consequently, when a nurse is assigned the off shift F1 on day *d*, there is only certain pairs of shifts allowed on day *d*-1 and *d*+1 such that a 35-hour consecutive off period is obtained. On the contrary, the off shift “F” has no restriction as to what work shift are assigned the day before and after. If a nurse does not work weekend, the F1-shift is assigned to Sunday the same week. If the nurse is working weekend, the F1-shift should be assigned as close as possible to the weekend, preferably Thursday or Friday. Furthermore, after a work weekend, the nurse should be assigned an F-shift on either Monday or Tuesday the following week.

There must be a minimum of 11 hours rest between two shifts, e.g., a nurse cannot work a day shift followed by a night shift the same day, or a night shift followed by an evening shift the following day. However, an exception is made for the rest time between an E-shift and a D-shift. This rest time must be a minimum of 9 hours. This implies that a nurse may work an E-shift on day *d* and a D-shift on day *d*+1. However, it is preferable to minimize the E-D combination of shifts on consecutive days.

Furthermore, night shifts should always be followed by another night shift or a “sleep” day then an off day. A sleep day means the nurse cannot be assigned to work a shift the day the night shift ends. This is, however, already implied with the minimum of 11 hours of rest between two shifts. However, the nurse should also be assigned a free shift the day after the sleep day.

Balance of Workload

Daily working hours should not exceed 10 hours, and the total maximum per week is 54 hours. It is further preferable to avoid working the same shift type on several consecutive days.

4.1.5 Fairness

Some scheduling rules are implemented to facilitate a certain degree of fairness. Nurses with the same position percentage should have an equal number of night and evening shifts during the weekdays. Nurses with a higher position percentage should have either the same amount or a higher number of night and evening shifts compared to someone with a smaller position percentage. During weekends, day and evening shifts should be distributed as equally as possible.

4.2 Cyclical Schedule

Holidays and vacations are not accounted for in this schedule. All nurses work every third weekend, and for those who are eligible, one night shift weekend is assigned every nineth weekend. However, nurses who only work night shifts should work nights every work weekend. When the number of weeks in the planning horizon are passed, the schedule is repeated. Consequently, when creating the schedule, it is important that all scheduling rules are implemented in such a way that the last week is followed by week 1. E.g., if a nurse works weekend in the last week, a free day must follow on either Monday or Tuesday in week 1.

4.3 Calendar-Based Schedule

The calendar-based schedule problem usually involves a longer planning horizon. Furthermore, holidays and vacations are accounted for, causing a lot of additional consideration such that more scheduling rules must be implemented.

Vacations

In general, employees with a position percentage above 20% are assigned five weeks of vacation over a year. One week during winter or spring time, three weeks during summertime, and finally one week during the autumn. Normally, most nurses have a preference as to when they are assigned vacation, however this problem does not cover personal vacation requests. Nevertheless, to replicate a realistic situation, we will accommodate for a common circumstance. Oftentimes, nurses who have children in school wish to have their autumn and winter vacation in the same weeks as the schools. Hence, we impose a scheduling rule that says a minimum number of nurses must have vacation in those two specific weeks. Otherwise, autumn and winter vacations is distributed evenly to avoid disturbing the available workforce too much.

There are three possible week intervals where a summer vacation can be allocated. Since a substantial portion of the available work force is reduced during the summer period, the manpower plan can be slightly adapted. This incurs that a shortage of one nurse per day in total over all shift is allowed, but should be avoided if possible. Additional nurses can be temporarily employed to cover some of the missing workforce, which we refer to as summer nurses. Furthermore, it is assumed that the weekend nurses in the case are students who wish to work extra during the summer. Therefore, the weekend nurse´s position is changed to full-time for a predetermined number of consecutive weeks during the summer period. In this time, the weekend nurses can work any shift including night shifts.

Holidays

A nurse can be assigned to work a maximum number of holidays that falls on a weekday during the planning period. Holidays that fall on Saturday or Sunday are not considered holidays. A new off shift is introduced alongside the holidays, the F3-shift. If a nurse is not allocated a work shift on a holiday, they are assigned the F3-shift. However, this rule does not apply for weekend nurses. The F3-shift adds a specific number of hours into the calculation of total work hours. The number of hours added depends on the work position of the nurse. Furthermore, the demand on a holiday is the same as a Sunday.

Weekend Rotation

The nurses work approximately every third weekend, and there must always be at least two weeks off between work weekends. Night weekends can be assigned maximum every nineth weekend, except for night shift nurses. The summer nurses must have at least one weekend off between work weekends in the period they are working.

4.4 Summary List of Scheduling Rules

In this section we provide a compressed list of the scheduling rules explained previously in this chapter.

4.4.1 Common Scheduling Rules

This section presents a list of scheduling rules which will be the basis for both the cyclical and the yearly scheduling problem.

Table 4.1: Common scheduling rules

Common Scheduling Rules

Employees and Work Hours

- Scheduled work hours cannot exceed contracted work hours
- Some nurses are limited to when and what shifts they are allowed to be assigned to.
- A group of nurses only work during the weekend, referred to as «weekend nurses»
- A group of nurses only work night shifts, referred to as «night shift nurses»

Shifts and Demand

- All nurses must be assigned one shift every day, which can be either a work shift or an off shift.
- The manpower plan must be covered

Weekends

- The nurses either has the weekend off or work both Saturday and Sunday
- E-shifts on Fridays can only be allocated to nurses working the following weekend.
- A maximum number of weekend nurses can be assigned to work the same weekend but cannot be assigned the same shift.

Regulations and policies

- There should be a minimum of 11 hours rest between two shifts, with one exception; rest time between E-shift and D-shift must be a minimum of 9 hours.
- For each week, a continuous free period of at least 35 hours must be planned which extends over a full day. In Norway, this is commonly referred to as the F1 day.
- If a nurse works weekend, the F1 day should preferably be allocated on a day as close as possible to the weekend. Otherwise, the F1 days are designated to Sundays.
- The combination of E-shift and D-shift respectively on consecutive days should be avoided.
- N-shifts should always be followed by another night shifts or two off days.
- Daily working hours should not exceed 10 hours
- A nurse can work maximum 54 hours per week

Fairness

- Nurses with the same position percentage should have an equal number of night and evening shifts during the weekdays
- Nurses with a higher position percentage should have either the same amount or a higher number of night and evening shifts compared to someone with a smaller position percentage.
- During weekends, day and evening shifts should be distributed as equally as possible.

4.4.2 Cyclical Scheduling rules

Table 4.2*: Cyclical scheduling rules*

Cyclical Scheduling Rules
• All nurses work every third weekend • Night shift work weekends must be assigned every nineth weekend, except for night shift nurses. • Scheduling rules must be respected in the transition from the last week to first week

4.4.3 Calendar-Based Scheduling rules

Table 4.3: *Calendar-based scheduling rules*

Calendar-Based Scheduling Rules
Vacations
• Employees with a position percentage above 20% are assigned five weeks of vacation over a year • A minimum number of nurses must have vacation in the same week as the schools during autumn and winter. • There are three possible week intervals where a summer vacation can be allocated • A shortage in demand of one nurse per day in total over all shift is allowed during summer vacation period, but should be avoided • Additional nurses can be temporarily employed during the summer, referred to as «summer nurses» • The weekend nurse´s position is changed to full-time for a predetermined number of consecutive weeks during the summer period, in this time, the weekend nurses can work any shift including night shifts. • To ensure a certain experience level, at least one nurse who is not a weekend or summer nurse must be assigned to each shift in the summer period.
Holidays
• Nurses can be assigned to work a maximum number of holidays that falls on a weekday • Demand for nurses on a holiday is the same as a Sunday.
Weekends
• There must always be at least two weeks off between work weekends. • Night weekends can be assigned maximum every nineth week, except for night shift nurses.

5. Optimization Models

5.1 Introduction of Models

The nurse scheduling problems are formulated as mathematical MILP models, or mixed integer linear problems. This means that the objective functions are linear and subject to linear constraints. They are mixed problems since they contain a mix of continuous and discrete, or integer, variables. The following sections present the mathematical formulations of each of the problems.

5.2 Cyclical Schedule

In this section, the model formulation for the cyclical schedule is presented. The purpose of this model is to create a systematic cyclical schedule over a predetermined number of weeks. Since the schedule is repeated, it does not consider vacations or holidays. Usually, a cyclical schedule needs assisting schedules to account for special times of the year, for instance during the summer and Christmas time.

5.2.1 Sets

The problem calls for a wide range of sets within different categories. The sets are split into sets regarding nurses, shift types and days and weeks. Below is an overview of all sets used in the cyclical model. These are also used in the calendar-based model.

Table 5.1: *Model sets regarding nurses*

Sets regarding nurses	
$\mathcal{N}$	All nurses
$\mathcal{N}^N \subseteq \mathcal{N}$	Night shift nurses
$\mathcal{N}^W \subseteq \mathcal{N}$	Weekend nurses
$\mathcal{N}^E \subseteq \mathcal{N}$	Evening nurses
$\mathcal{N}^{noN} \subseteq \mathcal{N}$	Nurses who should not work any N-shifts
$\mathcal{N}^{noNW} \subseteq \mathcal{N}$	Nurses who should not work any N-shift weekends
$\mathcal{N}^{moreE} \subseteq \mathcal{N}$	Nurses who should have more E-shifts than D-shifts during weekdays

$\mathcal{N}^{eq} \subseteq \mathcal{N}$	Set of nurses who should have equal number of E- and N-shifts during weekdays
$\mathcal{N}^{80+} \subseteq \mathcal{N}$	Nurses working in 80% position or more, with no limitations regarding shift allocation
$\mathcal{N}^{100} \subseteq \mathcal{N}$	Nurses working in a 100% position
$\mathcal{N}^{80} \subseteq \mathcal{N}$	Nurses working in an 80% position

Table 5.2: *Model sets regarding shifts*

Sets regarding shifts	
$\mathcal{S}$	All shifts $\{D, E, N, F, F1\}$
$\mathcal{S}^W \subseteq \mathcal{S}$	Work shifts $\{D, E, N\}$
$\mathcal{S}^O \subseteq \mathcal{S}$	Off shifts $\{F, F1\}$
$\mathcal{P}^I \subseteq \mathcal{S}^W \times \mathcal{S}^W$	Incompatible work shift pairs (s_1, s_2) on consecutive days
$\mathcal{P}^U \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{S} \times \mathcal{S}$	Unwanted shift patterns (s_1, s_2, s_3, s_4) on consecutive days

Table 5.3: *Model sets regarding days and weeks*

Sets regarding days and weeks	
$\mathcal{W}$	All weeks $\{1 \ldots Q^W\}$
$\mathcal{D}$	All days $\{1 \ldots Q^D\}$
$\mathcal{D}_w \subseteq \mathcal{D}$	Days d in week w, $\bigcup_{w \in \mathcal{W}} \mathcal{D}_w = \mathcal{D}$
$\mathcal{D}^I \subseteq \mathcal{D}$	Days in weekends
$\mathcal{D}^{II} \subseteq \mathcal{D}$	Weekdays, $\mathcal{D}^I \cup \mathcal{D}^{II} = \mathcal{D}$
$\mathcal{D}^{III} \subseteq \mathcal{D}$	Weekdays excluding Fridays
$\mathcal{D}_w^W \subseteq \mathcal{D}$	Saturday and Sunday (weekend days) of week *w*
$\mathcal{D}^{Sun} \subseteq \mathcal{D}$	All Sundays
$\mathcal{D}_w^{Sun} \subseteq \mathcal{D}$	Sunday in week *w*, $\bigcup_{w \in \mathcal{W}} \mathcal{D}_w^{Sun} = \mathcal{D}^{Sun}$
$\mathcal{L}$	Label days of the week $\{Monday, Tuesday, \ldots, Saturday, Sunday\}$

$\mathcal{D}_l \subseteq \mathcal{D}$	Days d in day of the week l,	$\bigcup_{l \in \mathcal{L}} \mathcal{D}_l = \mathcal{D}$

5.2.2 Parameters

Some of the parameters in this model are created merely for the purpose of making the final model more flexible and easier to alter. Such parameters are shown below. Examples are Q^D and Q^W. They are numerical parameters containing the number of days and weeks in the model. By utilizing the parameter throughout the constraints, the time horizon of the model can easily be altered by changing the value of the parameters in the data file.

***Table 5.4**: Model limit parameters*

Parameters	
Q^D	Number of days in the planning period
Q^W	Number of weeks in the planning period
Q^{WKD}	Number of weekends each nurse should work
Q^{NWKD}	Number of night weekends each nurse should work
Q^{HSM}	Number of hours from a Sunday night shift that occurs the following Monday
max^H	Maximum number of work hours allowed per nurse per week
max^{NE}	Maximum number of only E- or only N-shifts in a row
max^{WS}	Maximum number of work shifts per nurse per week
max^{EN}	Maximum number of E- and N-shifts in total per nurse per week
max^{ST}	Maximum number of each shift type per nurse per week
max^z	Maximum deviation between contracted hours and scheduled hours for each nurse

Other parameters in the model differ based on the nurse, shift or day in question, as shown below. The slightly more complex parameter B_{s_1,s_2} relates to the F1 day and was included in the thesis after being inspired by the work of Beckmann & Klyve (2016), where a similar parameter was used. The use of this parameter will be further explained in chapter 5.2.5.

Table 5.5: *Model general parameters*

Parameters	
$A_{s,l}$	Minimum number of nurses to be assigned shift $s \in \mathcal{S}^W$ on day of the week $l \in \mathcal{L}$
H_s	Hours per shift $s \in \mathcal{S}^W$
E_n	Contracted average hours for nurse $n \in \mathcal{N}$
B_{s_1,s_2}	$\begin{cases} 1, & if\ shift\ s_1\ and\ s_2\ can\ NOT\ be\ assigned\ before\ and\ after\ an\ F1\ day \\ 0, & otherwise \end{cases}$

5.2.3 Decision Variables

The model formulation contains a mix of continuous, integer and binary decision variables, as seen below.

Table 5.6: *Model Decision Variables*

Variables	
$\boldsymbol{y}_n \geq 0$	Scheduled average hours for nurse *n*
$\boldsymbol{z}_n \geq 0$	Difference between contracted hours and scheduled hours for nurse *n*
$\boldsymbol{x}_{n,s,d}$	$\begin{cases} 1, & if\ nurse\ n\ is\ allocated\ shift\ s\ on\ day\ d \\ 0, & otherwise \end{cases}$
$\boldsymbol{h}_{n,w}$	$\begin{cases} 1, & if\ nurse\ n\ works\ the\ weekend\ in\ week\ w \\ 0, & otherwise \end{cases}$
$\boldsymbol{hn}_{n,w}$	$\begin{cases} 1, & if\ nurse\ n\ works\ night\ the\ weekend\ in\ week\ w \\ 0, & otherwise \end{cases}$
$\boldsymbol{ed}_{n,d}$	$\begin{cases} 1, & if\ nurse\ n\ works\ E\ shift\ on\ day\ d\ and\ D\ shift\ on\ day\ d+1 \\ 0, & otherwise \end{cases}$
$\boldsymbol{shift}_{n,s,w}$	$\begin{cases} 1, & if\ nurse\ n\ works\ at\ least\ one\ s\ shift\ during\ week\ w \\ 0, & otherwise \end{cases}$
$\boldsymbol{\delta}_n^+ \geq 0$	$\begin{cases} > 0, & if\ the\ difference\ in\ "D"\ and\ "E"\ shifts\ during\ weekend\ is\ > 0\ for\ nurse\ n \\ 0, & otherwise \end{cases}$

$$\boldsymbol{\delta}_n^- \leq 0 \qquad \begin{cases} > 0, & if\ the\ difference\ in\ "D"\ and\ "E"\ shifts\ during\ weekend\ is\ < 0\ for\ nurse\ n \\ 0, & otherwise \end{cases}$$

5.2.4 Objective Function

The objective function seeks to minimize three different measures. First, it minimizes $\boldsymbol{z}_n$ for weekend and night shift nurses. $\boldsymbol{z}_n$ is the deviation between the hours a nurse should work throughout the schedule and the hours they are assigned by the model. Due to the length of the different shifts and other hard constraints, it is not possible to create a plan where each nurse works exactly the number of hours that they should. The goal is to create a schedule where the deviation is less than the length of one shift for each nurse. After the model is complete, the remaining hours would be manually assigned in a way that does not breach any constraints. This task is not part of the scope of this thesis. The model can make the deviation lower than the length of one shift for most nurses, but for weekend and night nurses it is not possible. For these nurses, the deviation is instead minimized through the objective function. Equations (5.1) and (5.2) show how $\boldsymbol{y}_n$ and $\boldsymbol{z}_n$ are calculated in the model.

$$\sum_{s \in \mathcal{S}} \sum_{d \in \mathcal{D}} (\boldsymbol{x}_{n,d,s} * H_s) = \boldsymbol{y}_n * Q^W, \qquad \forall n \in \mathcal{N} \tag{5.1}$$

$$E_n * Q^W = \boldsymbol{z}_n * \boldsymbol{y}_n * Q^W, \qquad \forall n \in \mathcal{N} \tag{5.2}$$

The second part of the objective function minimizes the number of times nurses work an E-shift followed by a D-shift the next day. This is in other words allowed in the plan but should be avoided as much as possible, since this combination of shifts only allows for a 9-hour rest period. Norwegian sleep scientist Bjørn Bjorvatn expressed through an article from 2017 that the combination of E- and following D-shift leads to a larger lack of sleep than night shifts and can be very detrimental to nurses' health and wellbeing in the long run (Frifagbevegelsen, 2019). Equation (5.3) shows how the parameter $\boldsymbol{ed}_{n,d}$ is calculated/defined.

$$\boldsymbol{x}_{n,s_1,d} + \boldsymbol{x}_{n,s_2,d+1} \leq \boldsymbol{ed}_{n,d} + 1, \qquad \forall n \in \mathcal{N}, d \in \mathcal{D}^{II} \mid s_1 = "E", s_2 = "D" \tag{5.3}$$

Lastly, our objective function contains the integer variables of δ_n^+ and δ_n^-. The purpose of this is to ensure a fair distribution of D- and E- shifts during weekends. Minimizing the difference between the number of D- and E- shifts on weekends for each individual, ensures that all nurses who can work all shift types have approximately the same amount of D-shifts as E-shifts during the weekends. δ_n^+ and δ_n^- are calculated through equation (5.4).

$$\delta_n^+ + \delta_n^- = \sum_{d \in \mathcal{D}^I} x_{n,s_1,d} - \sum_{d \in \mathcal{D}^I} x_{n,s_2,d}\,, \qquad \forall n \in \mathcal{N} \mid s_1 = "D", s_2 = "E" \tag{5.4}$$

The complete objective function for the cyclical model is defined by equation (5.5).

$$min \sum_{n \in \mathcal{N}^W \cup \mathcal{N}^N} \boldsymbol{z}_n + \sum_{n \in \mathcal{N}} \sum_{d \in \mathcal{D}} \boldsymbol{ed}_{n,d} + \sum_{n \in \mathcal{N}} \boldsymbol{\delta}_n^+ - \sum_{n \in \mathcal{N}} \boldsymbol{\delta}_n^- \tag{5.5}$$

5.2.5 Constraints

5.2.5.1 Coverage Constraints

The plan created by the model should fulfill the demand of nurses for each shift. This number varies between shift types and days of the week and is formulated as the parameter $A_{s,l}$. Constraint (5.6) allows up to one extra nurse on each D-shift. This solution allows for both demand and required work hours to be met while evenly distributing the extra nurses throughout, so that there is only ever one extra nurse on a shift. It is not desirable for nurses to work more evening and night shifts than necessary, so for these shifts, the number of nurses should equal the demand. This is ensured by constraint (5.7).

$$A_{s,l} \leq \sum_{n \in \mathcal{N}} \boldsymbol{x}_{n,s,d} \leq A_{s,l} + 1, \qquad \forall l \in \mathcal{L}, d \in \mathcal{D}_l \mid s = "D" \tag{5.6}$$

$$\sum_{n \in \mathcal{N}} \boldsymbol{x}_{n,s,d} = A_{s,l}, \qquad \forall\, l \in \mathcal{L}, d \in \mathcal{D}_l \mid s \neq "D" \tag{5.7}$$

5.2.5.2 Hours

No nurse should work more than max^H hours during any given week. The number of hours worked by a nurse each week is determined by the shifts worked in the given week. However, the work week starts at 00.00 on Monday. Q^{HSM} hours of the Sunday night shift occur during

Monday. Hence, if a nurse works Sunday night before a given week, Q^{HSM} hours must be added to the number of hours worked that week. Accordingly, if a nurse works Sunday night a given week, Q^{HSM} hours should be removed from the hours worked that week, as they are in fact part of Monday the following week. Constraint (5.8) ensures that the number of hours each nurse works each week is below the maximum limit, for all weeks except the first one. Constraint (5.9) is necessary to cover the transition from the last week of the plan to the first week.

$$\sum_{s \in \mathcal{S}} \sum_{d \in \mathcal{D}_w} \left(\boldsymbol{x}_{n,s,d} * H_s\right) - \sum_{d_1 \in \mathcal{D}_w^{Sun}} \left(\boldsymbol{x}_{n,s,d_1} * Q^{HSM}\right) + \sum_{d_2 \in \mathcal{D}_{w-1}^{Sun}} \left(\boldsymbol{x}_{n,s,d_2} * Q^{HSM}\right) \leq max^H, \quad \forall n \in \mathcal{N}, w \in W \mid w > 1, s = "N" \tag{5.8}$$

$$\sum_{s \in \mathcal{S}} \sum_{d \in \mathcal{D}_w \mid w=1} \left(\boldsymbol{x}_{n,s,d} * H_s\right) - \sum_{d_1 \in \mathcal{D}_w^{Sun} \mid w=1} \left(x_{n,s,d_1} * Q^{HSM}\right) + \sum_{d_2 \in \mathcal{D}_w^{Sun} \mid w=Q^W} \left(x_{n,s,d_2} * Q^{HSM}\right) \leq max^H, \quad \forall n \in \mathcal{N} \mid s = "N" \tag{5.9}$$

Constraint (5.10) is put in place to limit the deviation $\boldsymbol{z}_n$ for each nurse, except weekend and night shift nurses, to a maximum max^z, hours for the whole period.

$$\boldsymbol{z}_n \leq max^z, \qquad \forall n \in \mathcal{N} \mid n \notin \mathcal{N}^W \cup \mathcal{N}^N \tag{5.10}$$

5.2.5.3 Periods of Rest

Each nurse should have at least one 35-hour time-period off from work every week. In the model, this is marked as an F1-shift. To ensure this, constraint (5.11) is included in the model. The constraint is inspired by the solution of the previously mentioned master thesis written by Beckmann & Klyve (2016) but is but adjusted slightly to fit this model. To ensure one F1-shift per nurse per week, constraint (5.11) checks if the F1-shift for nurse *n* can be placed on a given day *d* based on the shifts of that nurse the day before and after. If nurse *n* has a given shift s_1 the day before, and a shift s_2 the day after, the parameter B_{s_1,s_2} determines whether there can be an F1-shift on day d or not. If B_{s_1,s_2} is equal to one, and nurse *n* works this combination of shifts the day before and after day *d*, there cannot be an F1-shift on day *d*.

$$\boldsymbol{x}_{n,s_1,d-2} + \boldsymbol{x}_{n,s,d-1} + \sum_{s_2 \in \mathcal{S}} \boldsymbol{x}_{n,s_2,d} * B_{s_1,s_2} \leq 2, \quad \forall n \in \mathcal{N}, s_1 \in \mathcal{S}, d \in \mathcal{D} \mid d > 2, s = "F1" \tag{5.11}$$

To ensure that each nurse has one F1-shift per week, constraint (5.12) is added to the model.

$$\sum_{d \in \mathcal{D}_w} \boldsymbol{x}_{n,s,d} = 1, \qquad \forall n \in \mathcal{N}, w \in \mathcal{W} \mid s = "F1" \tag{5.12}$$

If a nurse does not work a given weekend, the F1-shift for that nurse should be placed on that Sunday. However, if a nurse does work that weekend, their F1-shift should be placed on either Thursday or Friday that same week. These points are ensured through constraint (5.13) and (5.14) as seen below.

$$\boldsymbol{h}_{n,w} + \boldsymbol{x}_{n,s,d} = 1, \qquad \forall n \in \mathcal{N}, w \in W, d \in \mathcal{D}_w^{Sun} \mid s = "F1" \tag{5.13}$$

$$\sum_{s \in \mathcal{S}^W} \boldsymbol{x}_{n,s,d} - \boldsymbol{x}_{n,s_1,d-3} - \boldsymbol{x}_{n,s_1,d-2} = 0, \qquad \forall n \in \mathcal{N}, w \in W, d \in \mathcal{D}_w^{Sun} \mid s_1 = "F1" \tag{5.14}$$

In addition, a nurse that works a given weekend should have either the following Monday, Tuesday or both days off from work. This is ensured by constraint (5.15) for all weeks except the last one. Constraint (5.16) is added to ensure the same for the transition from the last weekend of the plan to the first Monday and Tuesday.

$$\sum_{s \in \mathcal{S}^W} \boldsymbol{x}_{n,s,d} + \sum_{s \in \mathcal{S}^W} \boldsymbol{x}_{n,s,d+1} + \sum_{s \in \mathcal{S}^W} \boldsymbol{x}_{n,s,d+2} \leq 2, \qquad \forall n \in \mathcal{N}, d \in \mathcal{D}^{Sun} \mid d < Q^D \tag{5.15}$$

$$\sum_{s \in \mathcal{S}^W} \boldsymbol{x}_{n,s,d_1} + \sum_{s \in \mathcal{S}^W} \boldsymbol{x}_{n,s,d_2} + \sum_{s \in \mathcal{S}^W} \boldsymbol{x}_{n,s,d_2+1} \leq 2, \qquad \forall n \in \mathcal{N} \mid d_1 = Q^D, d_2 = 1 \tag{5.16}$$

5.2.5.4 General Shift Constraints

Each nurse should have one shift each day of the schedule, either a work shift or an off shift. This is ensured through constraint (5.17). Furthermore, constraint (5.18) ensures that no nurse is given more than a maximum of max^{WS} work shifts per week. It is also desirable to avoid too many evening and night shifts in each week, so constraint (5.19) limits the sum of E- and N-shifts per nurse per week to max^{EN}. There is also a constraint (5.20) that limits the number of shifts of each shift type per nurse per week to max^{ST}, to ensure that shifts vary through the week. This constraint is valid for all nurses except night nurses.

$$\sum_{s \in \mathcal{S}} \boldsymbol{x}_{n,s,d} = 1, \qquad \forall n \in \mathcal{N}, d \in \mathcal{D} \tag{5.17}$$

$$\sum_{s \in \mathcal{S}^W} \sum_{d \in \mathcal{D}_w} \boldsymbol{x}_{n,s,d} \leq max^{WS}, \qquad \forall n \in \mathcal{N}, w \in \mathcal{W} \tag{5.18}$$

$$\sum_{d \in \mathcal{D} w} \boldsymbol{x}_{n,s_1,d} + \sum_{d \in \mathcal{D}_w} \boldsymbol{x}_{n,s_2,d} \leq max^{EN}, \qquad \forall n \in \mathcal{N}, w \in W \mid n \notin \mathcal{N}^N, s_1 = "E", s_2 = "N" \tag{5.19}$$

$$\sum_{d \in \mathcal{D}_w} \boldsymbol{x}_{n,s,d} \leq max^{ST}, \qquad \forall n \in \mathcal{N}, s \in \mathcal{S}^W, w \in W \mid n \notin \mathcal{N}^N \tag{5.20}$$

5.2.5.5 Unwanted combinations of shifts

There are rules regarding minimum hours of rest between two work shifts for all the nurses. This means that some combinations of shifts on two consecutive days should not be allowed, as they do not meet the rest requirements. These combinations of shifts are put in the set $\mathcal{P}^I$. Constraints (5.21) and (5.22) ensure that the rest requirement between shifts is met.

$$\boldsymbol{x}_{n,s_1,d} + \boldsymbol{x}_{n,s_2,d+1} \leq 1, \qquad \forall n \in \mathcal{N}, (s_1, s_2) \in \mathcal{P}^I, d \in \mathcal{D} \mid d < Q^D \tag{5.21}$$

$$\boldsymbol{x}_{n,s_1,d_1} + \boldsymbol{x}_{n,s_2,d_2} \leq 1, \qquad \forall n \in \mathcal{N}, (s_1, s_2) \in \mathcal{P}^I \mid d_1 = Q^D, d_2 = 1 \tag{5.22}$$

In section 5.2.4 about the objective function, it is mentioned that going from an E-shift to a D-shift the next day is not desirable. The number of times this happens in the model is minimized through the objective function. In addition, constraint (5.23) ensures that this type of shift combination happens at most one time during the weekdays for each nurse during a given week.

$$\sum_{d \in \mathcal{D}_w} \boldsymbol{ed}_{n,d} \leq 1, \qquad \forall n \in \mathcal{N}, w \in W \tag{5.23}$$

There are also a few combinations of shifts that should not occur, as these combinations can be less desirable for nurses. The plan should avoid unnecessarily frequent changes between shift types. Constraint (5.24) ensures that there are no cases of the shift patterns D-E-D-E and E-D-E-D. Constraints (5.25)-(5.27) are put in place to secure the same for the transition from last days of the plan to the first days of the plan.

$$\boldsymbol{x}_{n,s_1,d} + \boldsymbol{x}_{n,s_2,d+1} + \boldsymbol{x}_{n,s_3,d+2} + \boldsymbol{x}_{n,s_4,d+3} \leq 3, \quad \forall n \in \mathcal{N}, (s_1, s_2, s_3, s_4) \in \mathcal{P}^U, d \in \mathcal{D} \mid d < Q^D - 2 \tag{5.24}$$

$$\boldsymbol{x}_{n,s_1,d_1-2} + \boldsymbol{x}_{n,s2,d_1-1} + \boldsymbol{x}_{n,s_3,d_1} + \boldsymbol{x}_{n,s_4,d_2} \leq 3, \quad \forall n \in \mathcal{N}, (s_1, s_2, s_3, s_4) \in \mathcal{P}^U \mid d_1 = Q^D, d_2 = 1 \tag{5.25}$$

$$\boldsymbol{x}_{n,s_1,d_1-1} + \boldsymbol{x}_{n,s_2,d_1} + \boldsymbol{x}_{n,s_3,d_2} + \boldsymbol{x}_{n,s_4,d_2+1} \leq 3, \quad \forall n \in \mathcal{N}, (s_1, s_2, s_3, s_4) \in \mathcal{P}^U \mid d_1 = Q^D, d_2 = 1 \tag{5.26}$$

$$\boldsymbol{x}_{n,s_1,d_1} + \boldsymbol{x}_{n,s_2,d_2} + \boldsymbol{x}_{n,s_3,d_2+1} + \boldsymbol{x}_{n,s_4,d_2+3} \leq 3, \quad \forall n \in \mathcal{N}, (s_1, s_2, s_3, s_4) \in \mathcal{P}^U \mid d_1 = Q^D, d_2 = 1 \tag{5.27}$$

Constraints (5.28) – (5.31) makes it so that no nurse, except night shift nurses, has more than a maximum of max^{NE} E- or N-shifts in a row.

$$\sum_{a=0}^{N^{\max}} \boldsymbol{x}_{n,s,d+a} \leq N^{\max}, \quad \forall n \in \mathcal{N}, s \in \mathcal{S}^W, d \in \mathcal{D} \mid d < Q^D - 2, s \neq "D", n \notin \mathcal{N}^N \tag{5.28}$$

$$\boldsymbol{x}_{n,s,d_1-2} + \boldsymbol{x}_{n,s,d_1-1} + \boldsymbol{x}_{n,s,d_1} + \boldsymbol{x}_{n,s,d_2} \leq max^{NE}, \quad \forall n \in \mathcal{N}, s \in \mathcal{S}^W \mid d_1 = Q^D, d_2 = 1, s \neq "D", n \notin \mathcal{N}^N \tag{5.29}$$

$$\boldsymbol{x}_{n,s,d_1-1} + \boldsymbol{x}_{n,s,d_1} + \boldsymbol{x}_{n,s,d_2} + \boldsymbol{x}_{n,s,d_2+1} \leq max^{NE}, \quad \forall n \in \mathcal{N}, s \in \mathcal{S}^W \mid d_1 = Q^D, d_2 = 1, s \neq "D", n \notin \mathcal{N}^N \tag{5.30}$$

$$\boldsymbol{x}_{n,s,d_1} + \boldsymbol{x}_{n,s,d_2} + \boldsymbol{x}_{n,s,d_2+1} + \boldsymbol{x}_{n,s,d_2+2} \leq max^{NE}, \quad \forall n \in \mathcal{N}, s \in \mathcal{S}^W \mid d_1 = Q^D, d_2 = 1, s \neq "D", n \notin \mathcal{N}^N \tag{5.31}$$

Evening and night shifts are considered inconvenient shifts. The finished schedule should avoid nurses, except night shift nurses, having whole weeks only consisting of inconvenient shifts. Equations (5.32) and (5.33) determine whether a nurse has a given shift type during a given week. If a nurse has both E- and N-shifts, constraint (5.34) ensures that the given nurse should also have at least one D-shift that same week. If a nurse has only E-shifts or only N-shifts, previously mentioned constraint (5.20) limits the number of each shift, so that is allowed in the model without needing a D-shift to balance it out.

$$\boldsymbol{shift}_{n,s,w} * 1000 \geq \sum_{d \in \mathcal{D}_w} \boldsymbol{x}_{n,s,d}\,, \qquad \forall n \in \mathcal{N}, s \in \mathcal{S}^W, w \in \mathcal{W} \tag{5.32}$$

$$\boldsymbol{shift}_{n,s,w} \leq \sum_{d \in \mathcal{D}_w} \boldsymbol{x}_{n,s,d}\,, \qquad \forall n \in \mathcal{N}, s \in \mathcal{S}^W, w \in \mathcal{W} \tag{5.33}$$

$$\boldsymbol{shift}_{n,s_1,w} + 1 \geq \boldsymbol{shift}_{n,s_2,w} + \boldsymbol{shift}_{n,s_3,w}, \quad \forall n \in \mathcal{N}, w \in \mathcal{W} \mid s_1 = "D", s_2 = "E", s_3 = "N" \tag{5.34}$$

5.2.5.6 Nurse Specific Constraints

Constraints (5.35) and (5.36) tells the model that weekend nurses should not work Monday through Thursday, and they should not work D-shifts on Fridays.

$$\boldsymbol{x}_{n,s,d} = 0, \qquad \forall n \in \mathcal{N}^W, s \in \mathcal{S}^W, d \in \mathcal{D}^{III} \tag{5.35}$$

$$\boldsymbol{x}_{n,s,d} = 0, \qquad \forall n \in \mathcal{N}^W, l \in \mathcal{L}, d \in \mathcal{D}_l \mid l = "Friday", s = "D" \tag{5.36}$$

Furthermore, night shift nurses should only work N-shifts, as determined in constraint (5.37).

$$\boldsymbol{x}_{n,s,d} = 0, \qquad \forall n \in \mathcal{N}^N, s \in \mathcal{S}^W, d \in \mathcal{D} \mid s \neq "N" \tag{5.37}$$

Constraint (5.38) ensures that nurses who should not work night shifts are not assigned any night shifts throughout the schedule.

$$\boldsymbol{x}_{n,s,d} = 0, \qquad \forall n \in \mathcal{N}^{noN}, d \in \mathcal{D} \mid s = "N" \tag{5.38}$$

In addition, nurses belonging to group N^{noNW} should not be given any N-shift weekends. This is ensured through constraint (5.39).

$$x_{n,s,d} = 0, \qquad \forall n \in \mathcal{N}^{noNW}, d \in \mathcal{D}^{I} \mid s = "N" \tag{5.39}$$

Some nurses should only be assigned to E-shifts. This is ensured through constraint (5.40).

$$\sum_{s \in \mathcal{S}^{W} | s \neq "E"} \sum_{d \in \mathcal{D}} x_{n,s,d} = 0, \qquad \forall n \in \mathcal{N}^{E} \tag{5.40}$$

In this model, there are also some nurses who should have more E-shifts than D-shifts on weekdays in total. Hence, constraint (5.41) is added to the model.

$$\sum_{d \in \mathcal{D}^{II}} x_{n,s_1,d} + 1 \leq \sum_{d \in \mathcal{D}^{II}} x_{n,s_2,d}\,, \qquad \forall n \in \mathcal{N}^{moreE} \mid s_1 = "D", s_2 = "E" \tag{5.41}$$

5.2.5.7 Night Shift Constraints

The night shift constraints ensure two things. The first group of constraints, (5.42), (5.43) and (5.44), make sure that an N-shift is always followed by either another N-shift or two days off, as mentioned in the problem description. This is a hard constraint stemming from Haukelands policy. This policy helps support good health among healthcare workers, giving them the time to catch up on sleep and turn their inner clock back around before returning to work.

$$x_{n,s_1,d} \leq x_{n,s_1,d+1} + \left(1 - \sum_{s \in \mathcal{S}^{W}} x_{n,s_1,d+2}\right), \qquad \forall n \in \mathcal{N}, d \in \mathcal{D} \mid d < Q^{D} - 1 \mid s_1 = "N" \tag{5.42}$$

$$x_{n,s_1,d_1-1} \leq x_{n,s_1,d_1} + \left(1 - \sum_{s \in \mathcal{S}^{W}} x_{n,s,d_2}\right), \qquad \forall n \in \mathcal{N} \mid s_1 = "N", d_1 = Q^{D}, d_2 = 1 \tag{5.43}$$

$$x_{n,s_1,d_1} \leq x_{n,s_1,d_2} + \left(1 - \sum_{s \in \mathcal{S}^{W}} x_{n,s,d_2+1}\right), \qquad \forall n \in \mathcal{N} \mid s_1 = "N", d_1 = Q^{D}, d_2 = 1 \tag{5.44}$$

The remaining night shift constraints, (5.45), (5.46) and (5.47), ensure that the plan avoids single N-shifts for night nurses, where the nurse has a day off before and after the N-shift. This helps even out the night shifts per week dealt to night nurses. As an example, if the night nurse should work four N-shifts within two weeks, they will not get one N-shift one week and three the following week, but instead two per week.

$$\sum_{s\in\mathcal{S}^F} \boldsymbol{x}_{n,s,d} + \boldsymbol{x}_{n,s_1,d+1} + \sum_{s\in\mathcal{S}^F} \boldsymbol{x}_{n,s,d+2} \leq 2, \qquad \forall n \in \mathcal{N}^N, d \in \mathcal{D} \mid d < Q^D - 1, s_1 = "N" \tag{5.45}$$

$$\sum_{s\in\mathcal{S}^F} \boldsymbol{x}_{n,s,d_1-1} + \boldsymbol{x}_{n,s_1,d_1} + \sum_{s\in\mathcal{S}^F} \boldsymbol{x}_{n,s,d_2} \leq 2, \qquad \forall n \in \mathcal{N}^N \mid d_1 = Q^D, d_2 = 1, s_1 = "N" \tag{5.46}$$

$$\sum_{s\in\mathcal{S}^F} \boldsymbol{x}_{n,s,d_1} + \boldsymbol{x}_{n,s_1,d_2} + \sum_{s\in\mathcal{S}^F} \boldsymbol{x}_{n,s,d_2+1} \leq 2, \qquad \forall n \in \mathcal{N}^N \mid d_1 = Q^D, d_2 = 1, s_1 = "N" \tag{5.47}$$

5.2.5.8 Weekend Constraints

Constraint (5.48) makes sure that if a nurse works the evening or night shift of a Friday, they work the following weekend.

$$\sum_{s\in\mathcal{S}^W} \boldsymbol{x}_{n,s,d} \geq \boldsymbol{x}_{n,s_1,d-2}, \qquad \forall n \in \mathcal{N}, s \in S^W, d \in \mathcal{D}^{Sun} \mid s_1 \neq "D" \tag{5.48}$$

If nurse *n* works a specific weekend *w*, the binary variable $\boldsymbol{h}_{n,w}$ should be equal to 1. Constraint (5.49) ensures this, while also making sure that a nurse works either both days of the weekend or none of them. Every nurse works every third weekend, as ensured by constraint (5.50). Constraint (5.51) makes sure that every nurse works Q^{WKD} weekends each.

$$\boldsymbol{h}_{n,w} = \sum_{s\in\mathcal{S}^W} \sum_{d\in\mathcal{D}_w^W} x_{n,s,d}\,, \qquad \forall n \in \mathcal{N}, w \in \mathcal{W} \tag{5.49}$$

$$\boldsymbol{h}_{n,w} = \boldsymbol{h}_{n,w+3}, \qquad \forall n \in \mathcal{N}, w \in \mathcal{W} \mid w < Q^D - 2 \tag{5.50}$$

$$\sum_{w\in\mathcal{W}} \boldsymbol{h}_{n,w} = Q^{WKD}, \qquad \forall n \in \mathcal{N} \tag{5.51}$$

If a nurse has N-shifts during a weekend, they should work the N-shift on Friday, Saturday, and Sunday. Each nurse, except for night shift nurses, should have Q^{NWKD} N-shift weekends throughout the schedule period. We get constraints (5.52) and (5.53).

$$\boldsymbol{hn}_{n,w} * 3 = \boldsymbol{x}_{n,s,d} + \boldsymbol{x}_{n,s,d-1} + \boldsymbol{x}_{n,s,d-2}, \qquad \forall n \in \mathcal{N}, w \in \mathcal{W}, d \in \mathcal{D}_w^{Sun} \mid s = "N" \tag{5.52}$$

$$\sum_{w \in \mathcal{W}} \boldsymbol{hn}_{n,w} = Q^{NWKD}, \qquad \forall n \in \mathcal{N}^{80+} \tag{5.53}$$

There should be a maximum of two weekend nurses working during the same weekend, and they should not work the same shifts. This is to always ensure a satisfactory number of experienced nurses or nurses in larger positions at work. Hence, the model includes constraints (5.54) and (5.55).

$$\sum_{n \in \mathcal{N}^W} \sum_{s \in \mathcal{S}^W} \sum_{d \in \mathcal{D}_w^W} \boldsymbol{x}_{n,s,d} \leq 2, \qquad w \in \mathcal{W} \tag{5.54}$$

$$\boldsymbol{x}_{n_1,s,d} + \boldsymbol{x}_{n_2,s,d} \leq 1, \qquad n_1 \in \mathcal{N}^W, n_2 \in \mathcal{N}^W, w \in \mathcal{W}, d \in \mathcal{D}_w^W, s \in \mathcal{S}^W \mid n_1 \neq n_2 \tag{5.55}$$

5.2.5.9 Fair Distribution of Shifts

The last category of constraints in the cyclical plan is constraints regarding a fair distribution of shifts. For weekdays, all 100% and 80% employees should have the same amount of E- and N-shifts as other nurses working the same percentage. The same is true for nurses in set $\mathcal{N}^{eq}$. This set contains nurses who work part time with a position of less than 80%. The equal distribution of E- and N-shifts is ensured by constraints (5.56), (5.57) and (5.58). At the same time, constraints (5.59) and (5.60) tells the model that the nurses working a higher percentage should have more E- and N-shifts than nurses working in a lower percentage

$$\sum_{d \in \mathcal{D}^{II}} \boldsymbol{x}_{n_1,s,d} = \sum_{d \in \mathcal{D}^{II}} \boldsymbol{x}_{n_2,s,d}, \qquad \forall n_1 \in \mathcal{N}^{100}, n_2 \in \mathcal{N}^{100}, s \in \mathcal{S}^W \mid s \neq "D", n_1 \neq n_2 \tag{5.56}$$

$$\sum_{d \in \mathcal{D}^{II}} \boldsymbol{x}_{n_1,s,d} = \sum_{d \in \mathcal{D}^{II}} \boldsymbol{x}_{n_2,s,d}, \qquad \forall n_1 \in \mathcal{N}^{80}, n_2 \in \mathcal{N}^{80}, s \in \mathcal{S}^W \mid s \neq "D", n_1 \neq n_2 \tag{5.57}$$

$$\sum_{d \in \mathcal{D}^{II}} \boldsymbol{x}_{n_1,s,d} = \sum_{d \in \mathcal{D}^{II}} \boldsymbol{x}_{n_2,s,d}, \qquad \forall n \in \mathcal{N}^{eq}, nn \in \mathcal{N}^{eq}, s \in \mathcal{S}^W \mid s \neq "D", n_1 \neq n_2 \tag{5.58}$$

$$\sum_{d \in \mathcal{D}^{II}} \boldsymbol{x}_{n_1,s,d} \geq \sum_{d \in \mathcal{D}^{II}} \boldsymbol{x}_{n_2,s,d}, \qquad \forall n_1 \in \mathcal{N}^{100}, n_2 \in \mathcal{N}^{80}, s \in \mathcal{S}^W \mid s \neq "D" \tag{5.59}$$

$$\sum_{d\in\mathcal{D}^{II}} \boldsymbol{x}_{n_1,s,d} \geq \sum_{d\in\mathcal{D}^{II}} \boldsymbol{x}_{n_2,s,d} + 1, \qquad \forall n_1 \in \mathcal{N}^{80}, n_2 \in \mathcal{N}^{eq}, s \in \mathcal{S}^W \mid s \neq "D" \tag{5.60}$$

5.3 Calendar-Based Plan

This section presents the model formulation for the calendar-based schedule. This form of schedule assigns shifts to specific dates, typically over a longer planning horizon than a cyclical schedule, taking vacations and holidays into account. This results in a more complex problem. To reduce the computational time, we decompose the problem into a two-stage optimization model. The stage-1 model assigns vacations and work weekends to the nurses and decides if the nurse works night in that weekend or not. A series of hard constraints must be respected, but generally this is a feasibility problem requiring little computational effort where multiple optimal solutions exists. The stage-2 model allocates shifts for the weekdays and weekends throughout the year, using the results from the stage-1 model as fixed inputs. No shifts are specifically assigned in the stage-1 model. However, when a nurse is predetermined to work night a specific weekend, assigning said nurse to the night shifts in that weekend is the only feasible option.

5.3.1 Stage-1 Model

In this section the stage 1 model is formulated. The purpose of this model is to assign vacation weeks and work weekends, as well as determining if the nurse is working night shift during a specific weekend or not.

5.3.1.1 Sets

The same sets regarding nurses defined in table 5.1 are still applicable to this problem. However, the set of nurses $\mathcal{N}$ is expanded to include several stand-in nurses who are temporarily employed during the summer period. This is to ensure coverage of demand. This group of nurses will be referred to as "summer nurses". Consequently, a new subset is defined:

Table 5.7: *Additional nurse sets in Calendar-Based Stage-1 Model*

Additional sets regarding nurses	
$\mathcal{N}^S \subseteq \mathcal{N}$	Summer nurses

We deemed it necessary to define in stage-1 what shifts are allowed to be allocated in the stage-2 model during the weekend. Consequently, we define two shift categories, one which represents night shift, and one which represents either a day or an evening shift. The reasoning behind this approach is that we need to ensure that night weekends are evenly distributed. Furthermore, not every nurse can work night shifts during the weekends, and we need to make sure that the results obtained from the stage-1 model will not make the stage-2 model infeasible. A new set regarding shift category is therefore defined for the stage-1 model.

Table 5.8: *Additional shift sets in Calendar-Based Stage-1 Model*

Sets regarding shifts	
$\mathcal{SC}$	Shift category $\{Night, NotNight\}$

The set of weeks $\mathcal{W}$ defined in table 5.3 are relevant for this model as well. However, because vacations are accounted for in the calendar-based plan, we need additional sets which represents possible vacation weeks and weeks summer nurses can be employed.

Table 5.9: *Additional week sets in Calendar-Based Stage-1 Model*

Additional sets regarding weeks	
$\mathcal{V}^S \subseteq \mathcal{W}$	Set of possible vacation weeks during the summer
$\mathcal{V}^{S2} \subseteq \mathcal{V}^S$	Set of possible start weeks for summer vacation period
$\mathcal{V}^W \subseteq \mathcal{W}$	Set of possible vacations weeks during the winter and spring period
$\mathcal{V}^A \subseteq \mathcal{W}$	Set of possible vacation weeks during autumn
$\mathcal{V}^{School} \subseteq \mathcal{W}$	Set of school vacation weeks
$\mathcal{SN} \subseteq \mathcal{V}^S \times \mathcal{V}^S$	Set of certain summer week pairs used to indicate the periods where a summer nurse can be employed

5.3.1.2 Parameters

The parameters Q^W and Q^{NWKD} defined in table 5.4 for the cyclical schedule will be included for this model as well. The following new parameters are defined for the stage-1 model.

***Table 5.10**: Additional parameters in Calendar-Based Stage-1 Model*

Additional parameters	
A_c	Minimum number of nurses to be assigned shift category $c \in \mathcal{SC}$
min_c^{EL}	The minimum number of nurses with a certain experience level required for shift category $c \in \mathcal{SC}$
$\mathrm{min}^{\mathrm{FS}}$	The minimum number of nurses assigned in each summer vacation period
max^{WN}	The maximum number of weekend nurses assigned to work the same weekend
$Q_w^{Vacation}$	The number of nurses to be assigned vacation the same week as the school's vacation weeks $w \in \mathcal{V}^{School}$.

5.3.1.3 Decision Variables

The decision variables $\boldsymbol{h}_{n,w}$ and $\boldsymbol{hn}_{n,w}$ defined for the cyclical schedule in table 5.6 are included in this model. The variable $\boldsymbol{x}_{n,s,w}$ defined in table 5.6 are included as well, but is adjusted to apply for shift category *c* instead of shift *s*. The adjusted $\boldsymbol{x}_{n,s,w}$ variable and new decision variables are defined below.

***Table 5.11:** Additional decision variables in Calendar-Based Stage-1 Model*

Additional decision Variables	
$\boldsymbol{x}_{n,c,w}$	$\begin{cases} 1, & if\ nurse\ n\ is\ assigned\ to\ work\ shift\ category\ c\ the\ weekend\ in\ week\ w \\ 0, & otherwise \end{cases}$
$\boldsymbol{nv}_{n,w}$	$\begin{cases} 1, & if\ nurse\ n\ is\ assigned\ to\ have\ vacation\ in\ week\ w \in \mathcal{W} \\ 0, & otherwise \end{cases}$

5.3.1.4 Objective Function

The objective function seeks to minimize any shortage of required workforce during the weekends in the summer period. Hence, a certain gap is allowed but will be penalized in the objective function. Some hard constraints are further included to control how and when a shortage is allowed, which will be presented in the next section.

$$Min \sum_{w \in \mathcal{V}^S} \sum_{c \in \mathcal{SC}} \left(A_c - \sum_{n \in \mathcal{N}} \boldsymbol{x}_{n,c,w} \right) \tag{5.61}$$

5.3.1.5 Constraints

General Constrains

The nurses are assigned to either work a weekend or not by constraint (5.62). If assigned to work, they will be allocated to the night shift category or the not night category. The variable $\boldsymbol{h}_{n,w}$ is equal to one if the nurse is assigned to work a weekend, regardless of shift category. On the other hand, $\boldsymbol{hn}_{n,w}$ is only equal to one if the nurse is assigned to the night shift category. This is ensured by constraint (5.63) and (5.64).

$$\sum_{c \in \mathcal{SC}} \boldsymbol{x}_{n,c,w} \leq 1, \qquad \forall\, n \in \mathcal{N}, w \in \mathcal{W} \tag{5.62}$$

$$\boldsymbol{h}_{n,w} = \sum_{c \in \mathcal{SC}} x_{n,c,w}\,, \qquad \forall\, n \in \mathcal{N}, w \in \mathcal{W} \tag{5.63}$$

$$\boldsymbol{hn}_{n,w} = x_{n,c,w}, \qquad \forall\, n \in \mathcal{N}, w \in \mathcal{W} \mid c = \text{"}Night\text{"} \tag{5.64}$$

Coverage Constraints

The total number of nurses assigned to each shift category must correspond to the demand. However, during the summer period, a substantial portion of the work force is reduced due to vacations. Consequently, a shortage of one nurse is allowed on a shift, however there cannot be a shortage on multiple shifts on the same day. Constraint (5.65) concerns the coverage of demand on all weekends except during the summer vacation period. Constraint (5.66) and (5.67) account for coverage of demand during the summer period. A minimum portion of the demand, min_c^{EL}, should be covered by nurses with a certain experience level, which is achieved through constraint (5.68)

$$\sum_{n \in \mathcal{N}} \boldsymbol{x}_{n,c,w} = A_c, \qquad \forall\, c \in \mathcal{SC}, w \in \mathcal{W} \mid w \notin \mathcal{V}^S \tag{5.65}$$

$$A_c - 1 \leq \sum_{n \in \mathcal{N}} \boldsymbol{x}_{n,c,w} \leq A_c, \qquad \forall c \in \mathcal{SC}, w \in \mathcal{V}^S \tag{5.66}$$

$$\sum_{c \in \mathcal{SC}} A_c - 1 \leq \sum_{n \in \mathcal{N}} \sum_{c \in \mathcal{SC}} \boldsymbol{x}_{n,c,w}\,, \qquad \forall w \in \mathcal{V}^S \tag{5.67}$$

$$\sum_{n \in \mathcal{N} | n \notin \mathcal{N}^W \cup \mathcal{N}^S} \boldsymbol{x}_{n,c,w} \geq min_c^{EL}, \qquad \forall\, c \in \mathcal{SC}, w \in \mathcal{W} \tag{5.68}$$

Work Weekends

How work weekends are distributed varies depending on different group of nurses and whether it is during the summer period or not. In general, with the exception of summer nurses, there must be a minimum of two weekends off between each work weekend. This is covered by constraint (5.69). Furthermore, constraint (5.70) ensures there is a minimum of 8 weekends between each night weekend for all nurses except the ones who are only working night shifts. A maximum number of weekend nurses can be assigned to work on the same weekend, which is ensured through constraint (5.71). This constraint does not apply during the summer period, as the main concern at that time is to ensure sufficient coverage of demand.

$$\sum_{a=0}^{2} \boldsymbol{h}_{n,w+a} \leq 1, \qquad \forall\, n \in \mathcal{N}, w \in \mathcal{W} | n \notin \mathcal{N}^S \tag{5.69}$$

$$\sum_{a=0}^{8} \boldsymbol{hn}_{n,w+a} \leq 1, \qquad \forall\, n \in \mathcal{N}, w \in \mathcal{W} | n \notin \mathcal{N}^N \tag{5.70}$$

$$\sum_{n \in \mathcal{N}^W} \boldsymbol{x}_{n,c,w} \leq max^{WN}, \qquad \forall c \in \mathcal{SC}, w \in \mathcal{W} | w \notin \mathcal{V}^S \tag{5.71}$$

The summer nurses are allowed to work maximum every other weekend, but there must be at least two weeks between each night weekend. This is captured in constraint (5.72) and (5.73). Further, we need to make sure the summer nurses are not assigned to work any weekend outside the summer period, which is addressed in constraint (5.74).

$$\boldsymbol{h}_{n,w} + \boldsymbol{h}_{n,w+1} \leq 1, \qquad \forall\, n \in \mathcal{N}^S, w \in \mathcal{V}^S \tag{5.72}$$

$$\sum_{a=0}^{2} \boldsymbol{hn}_{n,w+a} \leq 1, \qquad \forall\, n \in \mathcal{N}^S, w \in \mathcal{V}^S \tag{5.73}$$

$$\boldsymbol{h}_{n,w} = 0, \qquad \forall\, n \in \mathcal{N}^S, w \in \mathcal{W} | w \notin \mathcal{V}^S \tag{5.74}$$

To allow for weekend nurses to work a 100% position and thus night shifts during the summer, constraint (5.75) is included. However, there must be a minimum of 8 weekends between each night shift weekend. The weekend nurses have a lower position the rest of the year and are not assigned night shifts. This is ensured by constraint (5.76).

$$\sum_{a=0}^{8} \boldsymbol{hn}_{n,w+a} \leq 1, \qquad \forall n \in \mathcal{N}^{W}, w \in \mathcal{V}^{S} \tag{5.75}$$

$$\sum_{w \in \mathcal{W} | w \notin \mathcal{V}^{S}} \boldsymbol{hn}_{n,w} = 0, \qquad \forall\, n \in \mathcal{N}^{W} \tag{5.76}$$

Night Shifts

Constraint (5.77) and (5.78) are included to account for groups of nurses that are either restricted to only working night shifts, or not work night shifts at all. Constraint (5.79) ensures that a minimum number of night weekends are assigned to the relevant nurses. This constraint is included to evenly distribute night weekends.

$$\boldsymbol{x}_{n,c,w} = 0, \qquad \forall\, n \in \mathcal{N}^{N}, w \in \mathcal{W} \mid c = "NotNight" \tag{5.77}$$

$$\boldsymbol{x}_{n,c,w} = 0, \qquad \forall n \in \mathcal{N}^{noNW}, w \in \mathcal{W} \mid c = "Night" \tag{5.78}$$

$$\sum_{w \in \mathcal{W}} \boldsymbol{hn}_{n,w} \geq Q^{NWKD}, \qquad \forall\, n \in \mathcal{N}^{80+} \tag{5.79}$$

Vacations

A nurse cannot be assigned to have vacation and work the weekend in the same week. Furthermore, weekend nurses and summer nurses are not assigned vacation. This is covered by constraint (5.80) and (5.81).

$$\boldsymbol{nv}_{n,w} + \boldsymbol{h}_{n,w} \leq 1, \qquad \forall n \in \mathcal{N}, w \in \mathcal{W} \tag{5.80}$$

$$\boldsymbol{nv}_{n,w} = 0, \qquad \forall n \in \mathcal{N}^S \cup \mathcal{N}^W \tag{5.81}$$

The model is formulated to assign a continuous summer vacation period of three weeks. This is enforced in constraint (5.82) and will not apply for nurses who should not be assigned vacation during the summer e.g., summer nurses. There are three possible periods of 3 weeks which nurses can be assigned to have summer vacation. The set $\mathcal{V}^{SS}$ represents the first week in each period. To ensure the nurses is assigned correctly to a period, constraint (5.83) is included. Constraint (5.84) ensures an even number of nurses are allocated to the tree different summer vacation alternatives.

$$\boldsymbol{nv}_{n,w} * 2 = \boldsymbol{nv}_{n,w+1} + \boldsymbol{nv}_{n,w+2}, \qquad \forall\, n \in \mathcal{N}, w \in \mathcal{V}^{SS} \mid n \notin \mathcal{N}^S \cup \mathcal{N}^W \tag{5.82}$$

$$\sum_{w \in \mathcal{V}^{SS}} \boldsymbol{nv}_{n,w} = 1, \qquad \forall\, n \in \mathcal{N} \mid n \notin \mathcal{N}^S \cup \mathcal{N}^W \tag{5.83}$$

$$\sum_{n \in \mathcal{N} \mid n \notin \mathcal{N}^S \cup \mathcal{N}^W} \boldsymbol{nv}_{n,w} \geq min^{FS}, \qquad \forall\, w \in \mathcal{V}^{SS} \tag{5.84}$$

The model allocates one week vacation during winter and one week during the autumn. This is ensured by constraint (5.85) and (5.86).

$$\sum_{w \in \mathcal{V}^{W}} \boldsymbol{nv}_{n,w} = 1\,, \qquad \forall\, n \in \mathcal{N} \mid n \notin \mathcal{N}^S \cup \mathcal{N}^W \tag{5.85}$$

$$\sum_{w \in \mathcal{V}^{A}} \boldsymbol{nv}_{n,w} = 1\,, \qquad \forall\, n \in \mathcal{N} \mid n \notin \mathcal{N}^S \cup \mathcal{N}^W \tag{5.86}$$

Many nurses wish to have vacation at the same time as the schools. The model is therefore formulated to assign vacation to a certain number of nurses in those weeks, which is usually one week during the autumn and one week during the winter. Consequently, we get the constraint (5.87). Otherwise, the remaining vacation weeks are distributed evenly ensured by constraint (5.88).

$$\sum_{n \in \mathcal{N} \mid n \notin \mathcal{N}^S \cup \mathcal{N}^W} \boldsymbol{nv}_{n,w} = Q_w^{Vacation}\,, \qquad \forall\, w \in \mathcal{V}^{School} \tag{5.87}$$

$$1 \leq \sum_{n \in \mathcal{N} \mid n \notin \mathcal{N}^S \cup \mathcal{N}^W} \boldsymbol{nv}_{n,w} \leq 2, \qquad \forall\, w \in \mathcal{V}^A \cup \mathcal{V}^W | w \notin \mathcal{V}^{School} \tag{5.88}$$

5.3.2 Stage-2 Model

In this section we present the stage-2 model. The purpose of this model is to allocate nurses to specific shifts for the whole year. This model will use the variables regarding weekend assignments and vacation, $\boldsymbol{h}_{n,w}$, $\boldsymbol{hn}_{n,w}$ and $\boldsymbol{nv}_{n,w}$, decided in stage-1 as fixed input parameters. The stage-2 model is combining elements of the cyclical schedule and the stage 1-model. Consequently, the sets, parameters, decision variables and constraints defined earlier in chapter 5.2 and 5.3.1 are implemented in the stage-2 model as well. These will not be repeated in this section.

5.3.2.1 Sets

Additional sets are defined for the stage-2 model regarding holidays. Further, to allow the weekend nurses to increase the work position to 100%, a new set of pairs are defined. The set represent different week intervals where a weekend nurse can work full-time. The first element is the first week and the second element is the last week of the interval. This set is defined as $\mathcal{WN}$.

Table 5.12: *Additional sets in Calendar-Based Stage-2 Model*

Additional sets	
$\mathcal{D}^H \subseteq \mathcal{D}$	Public holidays
$\mathcal{D}^{BH} \subseteq \mathcal{D}$	Days before a public holiday
$\mathcal{D}^{SS} \subseteq \mathcal{D}$	Days during the summer season
$\mathcal{D}^{HSS} \subseteq \mathcal{D}$	Public holidays and days during the summer season, $\mathcal{D}^H \cup \mathcal{D}^{SS} = \mathcal{D}^{HSS}$
$\mathcal{WN} \subseteq \mathcal{V}^S \times \mathcal{V}^S$	Set of certain summer week pairs used to indicate the periods where a weekend nurse can work fulltime

5.3.2.2 Parameters

Additional parameters are introduced for the stage-2 model. The decision variables $\boldsymbol{h}_{n,w}$, $\boldsymbol{hn}_{n,w}$ and $\boldsymbol{nv}_{n,w}$ defined in stage-1 will be implemented as fixed parameters. The new parameters are defined below.

Table 5.13: *Additional parameters in Calendar-Based Stage-2 Model*

Additional Parameters	
Q^{WS}	Number of work weeks for summer nurses
Q^{WW}	Number of weeks the weekend nurses should work a 100% position
Q^{VW}	Total number of vacation weeks to be assigned
max^{HD}	Maximum times a nurse can be assigned to work on a holiday, not included night shift starting on a holiday
max^{HDN}	Maximum number of times a nurse can be assigned to work on a holiday + night shift starting on a holiday
P_n	Percentage of the position of nurse n
H_n	Input parameter from Stage-1 Model: Assigned work weekends for nurse n
HN_n	Input parameter from Stage 1 Model: Assigned N-shift weekends for nurse n
$NV_{n,w}$	Input parameter from Stage 1 Model: Assigned vacation weeks for nurse n

5.3.2.3 Decision Variables

Some new binary decision variables are defined for the stage-2 model.

Table 5.14: *Additional decision variables for Calendar-Based Stage-2 Model*

Additional Decision Variables	
$\boldsymbol{F3}^{O}_{n,d}$	$\begin{cases} 1, & if\ nurse\ n\ is\ allocated\ an\ off\ shift\ on\ day\ d \in \mathcal{D}^H \\ 0, & otherwise \end{cases}$
$\boldsymbol{F3}^{W}_{n,d}$	$\begin{cases} 1, & if\ nurse\ n\ is\ allocated\ a\ work\ shift\ on\ day\ d \in \mathcal{D}^H \\ 0, & otherwise \end{cases}$
$\boldsymbol{ft}_{n,w}$	$\begin{cases} 1, & if\ nurse\ n \in \mathcal{N}^W works\ a\ full\ time\ position\ in\ week\ w \\ 0, & otherwise \end{cases}$
$\boldsymbol{sw}_{n,w}$	$\begin{cases} 1, & if\ nurse\ n \in \mathcal{N}^S\ works\ in\ week\ w \\ 0, & otherwise \end{cases}$

$$\boldsymbol{lack}_{s,d} \quad \begin{cases} 1, & if\ there\ is\ a\ shortage\ of\ one\ nurse\ for\ shift\ s\ on\ day\ d \\ 0, & otherwise \end{cases}$$

5.3.2.4 Objective Function

The three first components of the objective functions are obtained from the objective function in the cyclical schedule model. These are the components regarding the measures of $\boldsymbol{ed}_{n,d}$, $\boldsymbol{\delta}_n^+$ and $\boldsymbol{\delta}_n^-$, and the interpretation of them are the same as in the cyclical schedule. Consequently, the computation of $\boldsymbol{ed}_{n,d}$, $\boldsymbol{\delta}_n^+$ and $\boldsymbol{\delta}_n^-$ is equivalent to equation (5.3), (5.4) and (5.6) respectively. The model is able to find a possible solution where the difference in contracted and scheduled hours are less than the normal length of a shift. The component regarding $\boldsymbol{z}_n$ is therefore omitted, which will also reduce the computational time. A new component (to be minimized) is added, $lack_{s,d}$, which is a binary variable indicating if there is a shortage of nurses on shift $s \in \mathcal{S}^W$ on day $d \in D^{SS}$. Equation (5.89) presents how $lack_{s,d}$ is calculated in the model.

$$\boldsymbol{lack}_{s,d} = A_{s,l} - \sum_{n \in \mathcal{N}} \boldsymbol{x}_{n,s,d}, \qquad \forall s \in \mathcal{S}^W, l \in \mathcal{L}, d \in \mathcal{D}_l \mid d \in \mathcal{D}^{SS} \tag{5.89}$$

The sum of nurses assigned to a shift is limited to be maximum one less than demand, $A_{s,l}$, through constraint (5.91) presented in the next section.

We get the following objective function for the calendar-based stage-2 model:

$$\min \sum_{n \in \mathcal{N}} \sum_{d \in \mathcal{D}} \boldsymbol{ed}_{n,d} + \sum_{n \in \mathcal{N}} \boldsymbol{\delta}_n^+ - \sum_{n \in \mathcal{N}} \boldsymbol{\delta}_n^- + \sum_{s \in \mathcal{S}} \sum_{d \in \mathcal{D}} \boldsymbol{lack}_{s,d} \tag{5.90}$$

5.3.2.5 Constraints

The stage-2 model incorporates some constraint defined in the cyclical schedule. Some are identical, and will therefore not be redefined, but we make a reference to the relevant constraints. Other constraints that needed adjustments and new constraints will be presented in this section.

Coverage Constraints

Demand constraints are slightly more complex in the calendar-based model compared to the cyclical model. Demand depends on shift type, day of the week and whether it is a normal day, day during the summer vacation weeks or a holiday. On normal days, one extra nurse is allowed on D-shifts. However, for E-shifts and N-shifts the number of nurses at work should be equal to demand, as seen below in constraint (5.91), (5.92) and (5.93).

$$A_{s,l} \leq \sum_{n \in \mathcal{N}} \boldsymbol{x}_{n,s,d} \leq A_{s,l} + 1, \qquad \forall\, l \in \mathcal{L}, d \in \mathcal{D}_l \mid d \notin \mathcal{D}^{HSS}, s = "D" \tag{5.91}$$

$$\sum_{n \in \mathcal{N}} \boldsymbol{x}_{n,s,d} = A_{s,l}, \qquad \forall\, l \in \mathcal{L}, d \in \mathcal{D}_l \mid d \notin \mathcal{D}^{HSS}, s = "E" \tag{5.92}$$

$$\sum_{n \in \mathcal{N}} \boldsymbol{x}_{n,s,d} = A_{s,l}, \qquad \forall\, l \in \mathcal{L}, d \in \mathcal{D}_l \mid d \notin \mathcal{D}^{HSS}, d \notin \mathcal{D}^{BH}, s = "N" \tag{5.93}$$

For days during the summer season, one extra nurse is still allowed during D-shifts, but not during E- and N-shifts. A shortage of one nurse is allowed for each shift in the summer period. However, there should not be a shortage for more than one shift during a day. This is ensured by constraint (5.94), (5.95) and (5.96).

$$A_{s,l} - 1 \leq \sum_{n \in \mathcal{N}} \boldsymbol{x}_{n,s,d} \leq A_{s,l}, \qquad \forall s \in \mathcal{S}^W, l \in \mathcal{L}, d \in \mathcal{D}_l \mid d \in \mathcal{D}^{SS} \tag{5.94}$$

$$\sum_{s \in \mathcal{S}^W} \boldsymbol{lack}_{s,d} \leq 1, \qquad \forall d \in \mathcal{D}^{SS} \tag{5.96}$$

Holidays

Holidays have the same demand as Sundays for each shift type. In addition, an N-shift going into a holiday has the same demand as a Sunday N-shift. This is included in the model through the following constraints.

$$\sum_{n \in \mathcal{N}} \boldsymbol{x}_{n,s,d} = A_{s,l}, \qquad \forall\, s \in \mathcal{S}^W, d \in \mathcal{D}^H \mid l = "Sunday" \tag{5.97}$$

$$\sum_{n \in \mathcal{N}} \boldsymbol{x}_{n,s,d} = A_{s,l}, \qquad \forall d \in \mathcal{D}^{BH} \mid l = "Sunday", s = "N" \tag{5.98}$$

Hours

Calculation of work hours must be approached differently in the calendar-based schedule compared to the cyclical schedule to account for F3 shifts and vacations. For all nurses except weekend nurses and summer nurses, hours should be added to their work hours when they have a holiday off. The same is true for vacation weeks. These hours should be according to the percentage of their position, as seen in constraint (5.99). For weekends nurses, the calculation is presented in constraint (5.100). Work hours for summer nurses are calculated in a similar manner in constraint (5.101) and calculated as a weekly average based on the number of weeks they work.

$$\sum_{s \in \mathcal{S}, d \in \mathcal{D}} (\boldsymbol{x}_{n,d,s} * H_s) + 7.5 * P_n * \sum_{d \in \mathcal{D}^H} F3_{n,d} + 35.5 * P_n * Q^{VW} = \boldsymbol{y}_n * Q^W, \qquad \forall n \in \mathcal{N} \mid n \notin \mathcal{N}^W \cup \mathcal{N}^S \tag{5.99}$$

$$\sum_{s \in \mathcal{S}, d \in \mathcal{D}} (\boldsymbol{x}_{n,d,s} * H_s) = \boldsymbol{y}_n * Q^W, \qquad \forall n \in \mathcal{N}^W \tag{5.100}$$

$$\sum_{s \in \mathcal{S}, d \in \mathcal{D}^{SS}} (\boldsymbol{x}_{n,d,s} * H_s) = \boldsymbol{y}_n * Q^{WS}, \qquad \forall n \in \mathcal{N}^S \tag{5.101}$$

For most nurses, the difference in contracted hours and scheduled hours is calculated in constraint (5.102), similarly to equation (5.2) in the cyclical schedule. However, for the weekend and summer nurses, it is addressed differently. The hours a weekend nurse should work is adjusted to include the six weeks of full-time work during the summer in constraint (5.103). For summer nurses, the difference is calculated using Q^{WS} in constraint (5.104), which is the number of weeks the summer nurses work.

$$E_n * Q^W = \boldsymbol{z}_n * \boldsymbol{y}_n * Q^W, \qquad \forall n \in \mathcal{N} \mid n \notin \mathcal{N}^W \cup \mathcal{N}^S \tag{5.102}$$

$$E_n * (Q^W - Q^{WW}) + 35.5 * Q^{WW} = \boldsymbol{z}_n * \boldsymbol{y}_n * Q^W, \qquad \forall n \in \mathcal{N}^W \tag{5.103}$$

$$E_n * Q^{WS} = \boldsymbol{z}_n * \boldsymbol{y}_n * Q^{WS}, \qquad \forall n \in \mathcal{N}^S \tag{5.104}$$

Similarly to constraint (5.10) in the cyclical model, the difference in contracted and scheduled hours $\boldsymbol{z}_n$must be less than or equal to a certain number of hours $max^z,$. For the stage-2 model, this constraint applies to all nurses unlike the cyclical schedule.

$$\boldsymbol{z}_n \leq max^z, \qquad \forall n \in \mathcal{N} \tag{5.105}$$

Furthermore, to ensure a limit to how many hours a nurse can work in a week, the constraint (5.8) defined for the cyclical model is included in an identical manner for the stage-2 model.

Periods of rest

The constraints regarding rest time are identical to constraints (5.11) - (5.16) in the cyclical model. These constraints secure that each nurse has one F1-shift every week, as well as some time off before and after a work weekend.

General shift constraints

The general shift constraints are identical to the cyclical model, which refers to constraint (5.17) - (5.20.)

Unwanted combinations of shifts

Similarly, constraints regarding unwanted combinations of shifts are identical to constraint (5.21), (5.23), (5.24), (5.28) and (5.32) - (5.34) in the cyclical model. The constraints regarding the transition from the last week and first week are not necessary in this schedule because it is non-cyclical.

Nurse specific constraints

In addition to constraint (5.35) – (5.41) from the cyclical schedule, constraint (5.106) is included to ensure summer nurses are not assigned any shifts outside the summer period.

$$\sum_{s \in S^W} \boldsymbol{x}_{n,s,d} = 0, \qquad \forall\, n \in \mathcal{N}^S, d \in \mathcal{D} \mid d \notin \mathcal{D}^{SS} \tag{5.106}$$

Night shift constraints

The rules regarding night shifts are formulated identical to constraint (5.42) and (5.45) in the cyclical schedule.

Weekend Constraints

The first weekend constraints included in the stage 2 model are identical to constraint (5.48), (5.49) and (5.52) in the cyclical schedule. However, weekends are assigned in the stage-1 model. Consequently, the variables $\boldsymbol{h}_{n,w}$ and $\boldsymbol{hn}_{n,w}$ in constraint (5.49) and (5.52) are replaced by the fixed parameters $H_{n,w}$ and HN_{nw} . For this model, constraint (5.107) ensures that if parameter $H_{n,w}$ is equal to 1 then the given nurse must be assigned a work shift both Saturday and Sunday that weekend. Constraint (5.109) ensures that if a nurse is determined to work night during the weekend, the nurse is assigned the night shifts on Friday, Saturday and Sunday.

$$H_{n,w} = \sum_{s \in \mathcal{S}^W} \sum_{d \in \mathcal{D}_w^W} x_{n,s,d}\,, \qquad \forall n \in \mathcal{N}, w \in \mathcal{W} \tag{5.107}$$

$$HN_{n,w} * 3 = \boldsymbol{x}_{n,s,d} + \boldsymbol{x}_{n,s,d-1} + \boldsymbol{x}_{n,s,d-2}, \qquad \forall n \in \mathcal{N}, w \in \mathcal{W}, d \in \mathcal{D}_w^{Sun} \mid s = "N" \tag{5.110}$$

Furthermore, constraint (5.54) and (5.55) from the cyclical schedule is included in an identical manner as well.

Vacation Constraints

Constraint (5.111) ensures that if a nurse was assigned to have vacation in week *w* in the stage-1 model, the same nurse cannot be assigned to any shifts that week.

$$\boldsymbol{x}_{n,s,d} \leq 1 - NV_{n,w}, \qquad \forall n \in \mathcal{N}, s \in \mathcal{S}^W, w \in \mathcal{W}, d \in \mathcal{D}_w \mid n \notin \mathcal{N}^W \cup \mathcal{N}^S \tag{5.111}$$

Change of work position and Summer Nurses

To ensure weekend nurses do not work full time outside a part of the summer period, constraint (5.112) is included. Constraint (5.113) ensures that the weekend nurses work full-time a correct number of weeks. Because the weekend nurses can only work a particular number of weeks inside the summer period, constraint (5.114) ensures the full-time weeks are assigned consecutively. Constraint (5.115) ensures that if weekend nurse *n* is not assigned to work full-time a week during the summer period, nurse *n* should work as normal, only weekends. More specifically, nurse n cannot shifts from Monday-Thursday, D-shifts on Fridays or N-shifts during the weekend.

$$\sum_{w \in W | w \notin \mathcal{V}^S} \boldsymbol{ft}_{n,w} = 0, \qquad \forall n \in \mathcal{N}^W \tag{5.112}$$

$$\sum_{w \in \mathcal{V}^S} \boldsymbol{ft}_{n,w} = Q^{WW}, \qquad \forall n \in \mathcal{N}^W \tag{5.113}$$

$$\boldsymbol{ft}_{n,w_1} + \boldsymbol{ft}_{n,w_2} \leq 1, \qquad \forall n \in \mathcal{N}^W, (w_1, w_2) \in \mathcal{WN} \tag{5.114}$$

$$\begin{aligned} \boldsymbol{ft}_{n,w} * 1000 \geq & \sum_{s \in \mathcal{S}^W} \sum_{d \in \mathcal{D}_w | d \in \mathcal{D}^{III}} \boldsymbol{x}_{n,s,d} + \sum_{d \in \mathcal{D}_w | d \in \mathcal{D}_l | l = "Friday"} \boldsymbol{x}_{n,s_1,d} \\ & + \sum_{d \in \mathcal{D}_w^{Sun}} (\boldsymbol{x}_{n,s_2,d-2} + \boldsymbol{x}_{n,s_2,d-1} + \boldsymbol{x}_{n,s_2,d}), \qquad | \; s_1 = \mathrm{D}, s_2 = "N" \end{aligned} \tag{5.115}$$

Assignment of work weeks for the summer nurses are constructed in a similar way. Constraint (5.116) ensures the summer nurse works a certain number of weeks during the summer. Constraint (5.117) are included such that summer nurses cannot be assigned to weeks during the summer that they are not supposed to, similarly to constraint (5.115). Constraint (5.118) has the same purpose as constraint (5.114), the number of work weeks must be assigned consecutively. Constraint (5.119) make sure the summer nurse cannot be assigned a shift in a week they are not employed. Constraint (5.120) is included because it is preferable that temporarily full-time nurses are not alone on a shift, but should be accompanied by at least one of the nurses with a longer experience.

$$\sum_{w \in \mathcal{V}^S} \boldsymbol{sw}_{n,w} = Q^{WS}, \qquad \forall n \in \mathcal{N}^S \tag{5.116}$$

$$\boldsymbol{sw}_{n,w} * 1000 \geq \sum_{s \in \mathcal{S}^W} \sum_{d \in D_w} \boldsymbol{x}_{n,s,d}, \qquad \forall n \in \mathcal{N}^S, w \in \mathcal{V}^S \tag{5.117}$$

$$\boldsymbol{sw}_{n,w_1} + \boldsymbol{sw}_{n,w_2} \leq 1, \qquad \forall n \in \mathcal{N}^S, (w_1, w_2) \in \mathcal{SN} \tag{5.118}$$

$$\boldsymbol{x}_{n,s,d} \leq \boldsymbol{sw}_{n,w}, \qquad \forall\, n \in \mathcal{N}^S, s \in \mathcal{S}^W, w \in \mathcal{V}^S, d \in \mathcal{D}_w \tag{5.119}$$

$$\sum_{n \in \mathcal{N} | n \notin \mathcal{N}^W \cup \mathcal{N}^S} \boldsymbol{x}_{n,s,d} \geq 1, \qquad \forall s \in \mathcal{S}^W, d \in \mathcal{D}^{SS} \tag{5.120}$$

Holiday Constraints

Constraint (5.121) and (5.122) below are included to ensure a maximum limit to how many holidays a nurse can be assigned to work. Constraint (5.123) defines what shifts are considered to be on a holiday. Constraint (5.124) ensures that the decision variables indicating whether a nurse has been assigned to work on a holiday or not cannot be equal.

$$\sum_{d \in \mathcal{D}^H} \boldsymbol{F3}_{n,d}^W \leq \overline{M}^{HD}, \qquad \forall n \in \mathcal{N} \tag{5.121}$$

$$\sum_{d \in \mathcal{D}^H} \boldsymbol{F3}_{n,d}^W + \sum_{d \in \mathcal{D}^H} \boldsymbol{x}_{n,s,d} \leq \overline{M}^{HDN}, \qquad \forall n \in \mathcal{N} | s = "N" \tag{5.122}$$

$$\boldsymbol{F3}_{n,d}^W = \boldsymbol{x}_{n,s_1,d-1} + \sum_{s_2 \in \mathcal{S}^W | s_2 \neq "N"} \boldsymbol{x}_{n,s_2,d}\,, \qquad \forall n \in \mathcal{N}, d \in \mathcal{D}^H \mid n \notin \mathcal{N}^S, s_1 = "N" \tag{5.123}$$

$$\boldsymbol{F3}_{n,d}^W + \boldsymbol{F3}_{n,d}^O = 1, \qquad \forall n \in \mathcal{N}, d \in \mathcal{D}^H \mid n \notin \mathcal{N}^S \tag{5.124}$$

Fair distribution of shifts

Constraints regarding distribution of shifts are formulated identical to constraints (5.56) - (5.60) in the cyclical model.

6. Computational Implementation

The following section describes how the models' input data is created or determined, as well as the process of implementing the data in the optimization models and solving the models.

6.1 Data Description

Majority of the data used in the model comes from the case provided by staff at Haukeland University Hospital. Some data has been strictly predetermined in the case. Other data, however, has been determined through reasonable assumptions and considerations.

For the case implemented, the cyclical schedule is a nine-week schedule, while the calendar-based schedule runs over the course of 52 weeks. We get the following parameter values for each schedule:

	Parameter	Cyclical Schedule	Calendar-based Schedule
No. of Weeks	Q^W	9	52
No. of Days	Q^D	63	364
Plan Horizon		9 random weeks	March 6th 2023 – March 3rd 2024

The calendar-based plan runs from 6th of March 2023 until 3rd of March 2024. In other words, day 1 in the plan is the 6th of March 2023, and this is the first day of week 1. Based on this, we get the following sets of vacation weeks and holidays for the calendar-based plan.

	Set	Data	Description
Summer vacation	$\mathcal{V}^S$	$\{16 \dots 24\}$	Week
Start summer vacation	$\mathcal{V}^{S2}$	$\{16, 19, 22\}$	Week
Winter vacation	$\mathcal{V}^W$	$\{47 \dots 52, 1 \dots 7\}$	Week
Autumn vacation	$\mathcal{V}^A$	$\{27 \dots 39\}$	Week
School vacation	$\mathcal{V}^{School}$	$\{32, 52\}$	Week
Holidays	$\mathcal{D}^H$	$\{32, 33, 36, 57, 73, 74, 85, 295, 296, 302\}$	Day
Day before Holiday	$\mathcal{D}^{BH}$	$\{31, 32, 35, 56, 72, 73, 84, 294, 295, 301\}$	Day

The number of weeks in the plan determines parameters Q^{WKD} and Q^{NWKD}. For the cyclical plan, these parameters contain the number of weekends and night shift weekends a nurse should work. Parameter Q^{NWKD} is also included in the calendar-based plan, where it acts as a minimum number of night shift weekends assigned to some of the nurses. Parameter Q^{WKD} is calculated as the total number of weeks divided by 3, since all nurses should work every third weekend in the cyclical schedule. Parameter Q^{NWKD} is calculated as the number of weeks in the plan divided by 9 and rounded down when the result is not an integer number. We get the following values for the parameters:

Parameter	Cyclical Schedule	Calendar-based Schedule
Q^{WKD}	3	
Q^{NWKD}	1	5

Parameters Q^{WS} and Q^{WW} contain the number of weeks summer nurses should work and the number of weeks weekend nurses should work fulltime during the summer. Values for these parameters are determined through conversations with staff at Haukeland regarding reasonable assumptions for these parameters. Parameter Q^{VW} contains the number of vacation weeks each nurse should have over the course of 52 weeks. Nurses at Haukeland get five weeks of vacation. Parameter $Q_w^{Vacation}$ tells the model how many nurses should have vacation during the school vacation weeks, $\mathcal{V}^{School}$. This number is set to five in the implementation of the problem, as this was deemed a reasonable number by staff at the hospital. We get the following parameter values.

Parameter	Calendar-based Schedule
Q^{WS}	7
Q^{WW}	6
Q^{VW}	5
$Q_w^{Vacation}$	5

To ensure that the weeks summer nurses work during the summer, and the weeks weekend nurses work fulltime during the summer, are consecutive, sets $\mathcal{SN}$ and $\mathcal{WN}$ are introduced to the model. Both sets contain pairs of weeks during the summer. For each pair in $\mathcal{SN}$, a summer nurse can work at most one of the two weeks, as ensured through constraint (5.118). This ensures that each summer nurse works for 7 consecutive weeks during the summer. For each pair in $\mathcal{WN}$, a weekend nurse can work at most one of the two weeks, as ensured through constraint (5.114). This ensures that each weekend nurse works fulltime for 6 consecutive weeks during the summer. The two sets are listed below.

$set\ \mathcal{SN} := (16, 24)\ (16, 23)\ (17, 24)$

$set\ \mathcal{WN} := (16, 24)\ (16, 23)\ (16, 22)\ (17, 24)\ (17, 23)\ (18, 24)$

The data obtained from the case involves a mix of full-time and part-time nurses, with a total of 33 employees. All nurses are implemented in the model as a member of set $\mathcal{N}$, labelled $A1, A2, \dots, A33$, such that all sets regarding certain groups of nurses are a subset of $\mathcal{N}$. Each nurse has a contracted number of average weekly work hours stated in their contract, which corresponds to the parameter E_n in the models. Furthermore, the average work hours are often represented using the percentage of a full-time position. A full-time employee should work an average of 35.5 hours per week, which corresponds to a 100% position. The position percentage is needed in the calendar-based plan and corresponds to the parameter P_n. Some nurses are limited in terms of when and what shifts they should be assigned to. The table below presents an overview of the nurses, corresponding to set $\mathcal{N}$, their contracted work hours, E_n, the position percentage, P_n, and any limitations relevant for nurse *n*.

Table 6.1: *Data on nurse characteristics*

Nurses	P_n	E_n	***Limitations***
A1-A11	100%	35.500	
A12-A18	80%	28.400	
A19	60%	21.300	More evening shifts than day shifts in total during weekdays. Cannot work night shifts.
A20-A22	75%	26.625	Can only be assigned night shifts
A23	75%	26.625	Cannot work night shifts during weekends
A24	50%	17.750	Can only work evening shifts
A25-A27	50%	17.750	Cannot work night shifts during weekends
A28-A33	20%	7.100	Only works during weekends. Cannot work night shifts.

A nurse may work more than 35.5 hours in one week and fewer hours another week, but the maximum number of hours a nurse can work in a single week is given by the parameter max^H.

This parameter is set to 54 hours, since that is the maximum number of hours per week that is allowed based on regulations in the Norwegian Working Environment Act (Arbeidsmiljøloven, 2005, § 10-5). However, none of the nurses in this schedule get close to this limit, due to other constraints limiting the number of works shifts per week. It might still be useful to have this parameter in the model for other departments where the number of maximum shifts per week differ from our number.

In the calendar-based schedule the set regarding all nurses has 5 additional members, representing the temporarily employed summer nurses, which are labelled S1,…,S5. The addition of 5 summer nurses is implemented on the basis of obtaining a feasible solution. Any less than five would result in a bigger shortage of required workforce during the summer period than the model allows for. However, less summer nurses could be employed if a bigger shortage during the summer period was allowed. The required workforce during the summer period differs depending on the needs of the department in mind. The balance between under-coverage and number of summer nurses to employ is a difficult decision, and is to be determined by the ward manager. In our case we made the assumptioin, based on conversations with Haukeland, that under-coverage is weighted more heavily than minimizing the number of summer nurses employed.

The shifts in the data consists of three work shifts and two off shifts, labeled “D”, “E”, “N”, “F” and “F1” respectively. These are all members of set S. The general manpower plan for different weekdays and shift types for this case is presented in the table below.

Table 6.2: *Manpower plan for this case*

	H_s	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
D-shift 07:00am – 03:00pm	*8*	6	6	6	6	6	4	4
E-shift 02:30pm – 10:00pm	*7.5*	5	5	5	5	4	4	4
N-shift 09:30pm – 07:00am	*9.5*	4	4	4	4	3	3	3

The parameter H_s is derived from the information regarding shift lengths, provided by Haukeland, and the data for the parameter is presented in table 6.2. Parameter Q^{HSM}, which is the number of hours from a Sunday night shift which belong to the following week, is 7 based on the above shift times. The manpower plan corresponds to the data implemented in parameter $A_{s,l}$ in the cyclical plan and stage 2 of the calendar-based plan. For the stage-1 model, the parameter A_c is calculated based on the same numbers for weekend shifts

Shift Category ($s \in \mathcal{SC}$)	Value for parameter A_c	Value for parameter min^{EL}
Night	3	1
NotNight	8	2

The models also contain sets regarding incompatible shifts on consecutive days, as well as unwanted shift patterns. These sets are $\mathcal{P}^I$ and $\mathcal{P}^U$, respectively. Set $\mathcal{P}^I$ is determined by regulations regarding the minimum amount of rest time between two work shifts. As mentioned in the problem description, the rest period between two shifts should always be at least eleven hours long, except for E-shifts followed by a D-shift, where the rest period in this case is nine hours. To achieve this, the schedule should avoid assigning an N-shift to a nurse on day *d* with a D-shift or E-shift on day *d+1*. We get the following set $\mathcal{P}^I$:

$Set\ \mathcal{P}^I := ("N","D"), ("N","E")$

In addition, we wish to avoid unnecessary changes between shift types on consecutive days. Set $\mathcal{P}^U$ lists shift patterns that are not allowed in the model.

$Set\ \mathcal{P}^U := ("D","E","D","E"), ("E","D","E","D")$

In order to allocate F1-shifts, we use the parameter B_{s_1,s_2}. An F1-shift is 35 hours long and should consist of at least one full day off. In order for the model to allocate an F1-shift, there are some limitations regarding what combinations of shifts that can be worked on the day before and after the F1-shift. B_{s_1,s_2} is 1 for combinations of shifts that cannot happen, and 0 for the combinations of shifts that are okay. We get the following values for combinations of shifts (s_1, s_2) for parameter B_{s_1,s_2}.

	D	E	N	F	F1
D	0	0	0	0	0
E	1	0	0	0	0
N	1	1	1	0	0

F	0	0	0	0	0
F1	0	0	0	0	0

Figure 6.1: Parameter B_{s_1,s_2}

6.2 Implementation

The mathematical models are implemented and solved using the Gurobi solver in the mathematical modeling interface AMPL IDE. Three files are created for the purpose of solving the model for the cyclical schedule. These three files consist of one *.mod*-file, one *.dat*-file and one *.run*-file. The *.mod*-file contains the mathematical formulation of the problem, written in section 5 in this thesis. This includes an overview of sets, parameters, decision variables, objective function, and constraints. The *.dat*-file contains the data for all sets and parameters, and the *.run*-file combines the *.mod*- and *.dat*-file and solves the problem using the Gurobi solver. The Gurobi solver can solve linear and quadratic optimization in continuous and integer variables, so it suits our MILP-problems well (AMPL, 2022).

The calendar-based model is implemented and solved in a similar matter but contains two *.mod*-files and two *.dat*-files, to accommodate stage 1 and stage 2 of the problem. It is completed and ran through one common *.run*-file. The first step in the *.run*-file is to solve the stage-1 model using the associated *.mod*-file and *.dat*-file. Values for $h_{n,w}, hn_{n,w}$ and $nv_{n,w}$ are obtained from the solution. The next step in the *.run*-file is to store these variables. For this purpose, we designed an algorithm that transform the values into a data format that is readable to AMPL and imports them in a new *.dat*-file as parameters. The next step is then to solve the stage-2 model using the associated *.mod*-file and *.dat*-file as well as the new *.dat*-file generated in the previous step.

Once AMPL finds an optimal solution to the problem, the shifts assigned to each nurse for each day is exported to a *.txt*-file. We import this .txt-file to PyCharm and use the programming language Python to restructure the results and create a dataframe with the same format as Haukeland normally uses for their schedules. This dataframe is then exported to Excel and holds the same format as schedules at Haukeland.

7. Results

In this chapter we present the results from the optimization models using the data described in chapter 6.1. The main focus is how the models perform in terms of certain criteria, including computational time, fairness and coverage of demand. In the following sections we discuss the criteria further and present numerical results where applicable.

7.1 Final Schedule

The final schedules are formatted as seen in the excerpt in figure 7.1. The excerpt shows the first 4 weeks of the cyclical schedule created by the model.

	Week 1							Week 2							Week 3							Week 4						
	mon.	tue.	wed.	thu.	fri.	sat.	sun.	mon.	tue.	wed.	thu.	fri.	sat.	sun.	mon.	tue.	wed.	thu.	fri.	sat.	sun.	mon.	tue.	wed.	thu.	fri.	sat.	sun.
A1	D	E	E	E			F1	N	N		F1	E	D	E	E		D	D	D		F1	D	D	N	N			F1
A2	D	D	E	E			F1	D	D		F1	N	N	N			D	D	D		F1	D	E	E	N			F1
A3	E		D	D	D		F1	D	E	N	N			F1		D	E	F1	N	N	N			D	D	D		F1
A4			D	D	D		F1	E	E		D	D		F1	N	N		F1	E	D	E	E		D	D	D		F1
A5	E		D	D	D		F1	D	D	E	N			F1	N	N		F1	E	D	E	E		D	D	D		F1
A6	D	E	E	E			F1	E		D	F1	N	N	N			D	D	D		F1	D	E	N	N			F1
A7	N	N		F1	E	D	D	E		D	D	D		F1	D	E	N	N			F1	D	E	N	F1		E	E
A8			D	D	D		F1	E	E		D	D		F1	D	E	N		F1	E	D	E		D	D	D		F1
A9	E		D	D	D		F1	D	D	E	E			F1	D	N	N		F1	E	D	E		D	D	D		F1
A10	E		D	D	D		F1	D	E	N	N			F1		D	E	F1	N	N	N			D	D	D		F1
A11	D	D	N	N			F1	D	N		F1	D	E	E	E		D	D	D		F1	D	E	E	E			F1
A12		D		E	F1	E	E		N					F1		D	D	N			F1	N	N		F1	E	D	D
A13	D		E	F1	E	D	E		D	N				F1	E		E	E			F1		D		F1	N	N	N
A14	D	E		F1	N	N	N			D	E			F1	D	E		N			F1	N			E	F1	E	D
A15		E		E			F1		D		F1	E	D	D	E		D	D			F1	D	D	E	N			F1
A16	N	N		F1	E	D	D		D	E	E			F1	D	E	E	E			F1		D	D	F1	N	N	N
A17	D	D		F1	N	N	N			D	D	D		F1	E			D	D		F1	E			F1	E	D	D
A18		D	D	N			F1		D	D	E	F1	E	E		D	E	E			F1	D	N					F1
A19				D			F1	E		E	F1	E	D	D					D		F1		D	E				F1
A20			N	N			F1	N	N	N	N			F1	N	N		F1	N	N	N							F1
A21	N	N		F1	N	N	N	N						F1			N	N			F1	N	N		F1	N	N	N
A22	N	N	N	N			F1				F1	N	N	N	N						F1	N	N	N				F1
A23		D	E		D		F1			D	F1	D	E	D		D		D	D		F1		D		D	D		F1
A24		E			F1	E	E		E	E	E			F1				E			F1				E	F1	E	E
A25	E						F1	D			D			F1	D	E		F1		E	D		D	E	E			F1
A26			N		F1	E	D			D				F1		D		E			F1				F1	E	D	E
A27							F1	N			D			F1			D	F1	E	D	E		E		E			F1
A28							F1				F1		E	D							F1							F1
A29							F1				F1	E	D	E							F1							F1
A30							F1							F1				F1	E	D	D							F1
A31							F1							F1				F1		E	E							F1
A32				F1		E	E							F1							F1				F1	E	D	E
A33				F1	E	D	D							F1							F1				F1		E	D

Figure 7.1: *Excerpt from the final schedule for the cyclical model*

The calendar-based model incorporates vacations and holidays, as can be seen in the excerpt in figure 7.2. The excerpt displays the final plan for the month of July, where many nurses have their summer vacation. Vacations are marked with the letters "FE" and the color blue below. The excerpt also shows that summer nurses work during this period, and some of the weekend nurses work fulltime part of this month.

																	JULY														
	1.7.	2.7.	3.7.	4.7.	5.7.	6.7.	7.7.	8.7.	9.7.	10.7.	11.7.	12.7.	13.7.	14.7.	15.7.	16.7.	17.7.	18.7.	19.7.	20.7.	21.7.	22.7.	23.7.	24.7.	25.7.	26.7.	27.7.	28.7.	29.7.	30.7.	31.7.
						Week 18							Week 19							Week 20							Week 21				
	sat.	sun.	mon.	tue.	wed.	thu.	fri.	sat.	sun.	mon.	tue.	wed.	thu.	fri.	sat.	sun.	mon.	tue.	wed.	thu.	fri.	sat.	sun.	mon.	tue.	wed.	thu.	fri.	sat.	sun.	mon.
A1	E	E	E		D	D	D		F1	D	D	E	N			F1	E		D	D	D		F1	D	D		E	F1	E	D	
A2		F1		D	E	F1	N	N	N				FE							FE							FE				
A3						FE					D	D	E	F1	E	E		E	N	N			F1	D	D	E	N			F1	
A4		F1	D		D	F1	N	N	N			D	D	D		F1	D	D	E	N			F1	D	N	N	F1		E	E	
A5		F1	D	D	D	E			F1				FE							FE							FE				N
A6		F1	N	N		F1	D	E	E	E		D	D	D		F1	E	E	N	N			F1	N	N		F1	D	E	E	
A7	D	E	E		D	D	D		F1				FE							FE							FE				E
A8						FE				D	E	N	N			F1	N	N		F1	E	D	E	E		D	D	D		F1	D
A9	E	D		E	N	N			F1				FE							FE							FE				E
A10		F1	N	N		F1	E	D	E				FE							FE							FE				D
A11		F1	E	E	N	N			F1	D	E		F1	N	N	N			D	D	D		F1	E	E	E	N			F1	
A12						FE							F1	E	D	D		D	E	E			F1		E		D			F1	D
A13		F1	D	D	D				F1	E	N			D		F1	N			D	F1	E	D		E					F1	
A14						FE					D	N				F1		D		F1		E	E	N				D		F1	
A15						FE				E						F1				F1	N	N	N			N				F1	E
A16	E	E	D		N				F1	E	E		D			F1	E	E		D	F1	E	D	D			E			F1	
A17		F1	N	N			D		F1			E	F1	E	D	D			N	N			F1				D	D		F1	
A18	E	E		D		D	D		F1	N			E			F1		D			D		F1	D			D	F1	E	E	
A19		F1		E		D	F1	E	D				FE							FE							FE				
A20	N	N	N						F1				FE							FE							FE				N
A21						FE				N	N					F1	N	N					F1	N	N		F1	N	N	N	N
A22		F1							F1				FE							FE							FE				
A23		F1	D	D		D			F1			D	D	D		F1	D			E			F1	D	D		F1	E	D	E	
A24		F1		E		E			F1				FE							FE							FE				
A25						FE					E					F1				F1		D	E							F1	
A26						FE							F1		E	E	N						F1		D					F1	E
A27									F1				E	F1	E	E		N	N				F1	E						F1	
A28		F1				F1	E	E	D		N	N				F1	D	E	E	E			F1	E		D	F1	N	N	N	
A29	N	N			D	E			F1	D	D	E	N			F1	E		D	F1	E	E	D	E		N	N			F1	
A30	N	N			N	N			F1	D	D	E	E			F1	D	N		F1	E	D	E		E	E	E			F1	
A31		F1			E	E			F1	N			F1	E	D	E		D	D	E			F1	D	D	N	N			F1	
A32		F1							F1	E	N		F1	E	E	D		D			D		F1		E	E	E			F1	D
A33		F1				F1	E	D	E	N			D	D		F1	D	D	E	E			F1			D	F1	N	N	N	
S1	D	D	E		E	N			F1		D	E	F1	N	N	N			D	D	D		F1		D	E	F1	E	D	D	E
S2		F1		D	E	E	F1	E	D	D		N	N			F1	D		D	F1	N	N	N			D	D	D		F1	N
S3		F1	E	N		F1	E	D	D		D	D	E			F1	D		E	F1	N	N	N			D	D	D		F1	D
S4		F1	D		E	F1	N	N	N			D	D	D		F1	E	E		F1	E	D	D		D	D	E			F1	D
S5	D	D		E		D	D		F1	D	E		F1	N	N	N			D	D	D		F1	N	N		F1	E	D	D	

Figure 7.2: *Excerpt - summer weeks of the final schedule for the calendar-based model*

7.2 Computational Time

One of the main concerns of manually constructing work schedules is the considerable time required. However, with the models proposed in this thesis, the computational times are remarkably reduced. The run time is presented in the table below.

Table 7.1: *Model run times*

	Run time	***Optimality Gap %***
Cyclical Schedule	3350 seconds	0%
Calendar Schedule	6700 seconds	0%

The cyclical schedule model takes 3350 seconds, or a little less than one hour to find the optimal solution to the problem. Compared to manual methods at Haukeland which requires about 1-2 weeks, this is a considerable improvement. The manual process of creating a calendar-based schedule takes approximately 4-6 weeks. The run time for the calendar-based models is 6700 seconds, or a little less than two hours, in total of both stage-1 and stage-1. This can be considered a remarkable improvement. Implementing this model could therefore improve the utilization of human resources by freeing up a lot of time for the administrative staff involved in the scheduling process.

7.3 Fairness

In reality, it can be hard to measure fairness because a lot of factors are involved, and the perception of fair can differ between individuals. In this thesis, we have chosen to measure fairness based on how unpopular shift types and work weekends are allocated. If divided evenly, this would be considered a high degree of fairness. The unpopular shift types are E-shifts and N-shifts. Furthermore, working a weekend is generally undesirable, and working a night weekend even more so. The reader should note that N-shifts are naturally not considered unpopular for night shift nurses. The distribution of E-shifts, N-shifts, total number of work weekends and night weekends for the cyclical schedule are presented in table 7.2 below. Portion represents how many of the total number of nurses within the category was allocated the given numbers of each shift type and weekends.

Table 7.2: Distribution of unpopular shifts in Cyclical Schedule

		Weekdays		**Weekend**		**Work Weekends**	
Position	*Portion*	*E-shift*	*N-shift*	*E-shift*	*N-shift*	*Total*	*Night*
100%	11/11	9	7	2	2	3	1
80%	4/7	8	5	2	2	3	1
	3/7	8	5	1	2	3	1
75% Night Nurses	1/3	0	19	0	6	3	3
	2/3	0	18	0	6	3	3
75%	1/1	6	1	3	3	3	0
60%	1/1	13	0	3	3	3	0
50%	3/4	6	1	3	0	3	0
	1/4	15	0	6	0	3	0
20%	3/6	1	0	3	0	3	0
	3/6	2	0	3	0	3	0

Several hard constraints were included to ensure a certain degree of fairness. For this instance, these hard constraints did not make the problem infeasible, but it should be noted that with another data set this could potentially create problems. However, because of the hard constraints, the distribution of unpopular shifts during the weekdays are evenly distributed based on percentage position. The one with a 50% position with 15 E-shifts on weekdays and 6 E-shifts on the weekend refers to nurse A24, a nurse limited to only working E-shifts. The distribution of E-shifts and D-shifts during weekends was accounted for using soft constraints, and its violations were penalized in the objective function through the decision variables $\boldsymbol{\delta}_n^+$ and $\boldsymbol{\delta}_n^-$. Consequently, the distribution in the table above is the best possible solution based on how the model is formulated. With the exception of 80% nurses, E-shifts and N-shifts is distributed perfectly even within respective groups. For the 80% nurses, 4/7 works two E-shifts whereas the other 3/7 works one E-shift. This is still considered to be a fair distribution. Consequently, even distribution of E- and N-shifts during weekends are obtained to a satisfying degree. Work weekends and night weekends are distributed equally within the respective groups. On the basis of these results, the cyclical schedule model performs to a high degree regarding fairness.

Table 7.3: *Distribution of unpopular shifts in Calendar-Based schedule*

		Weekdays		**Weekend**		**Work Weekends**	
Position	*Portion*	*E-shift*	*N-shift*	*E-shift*	*N-shift*	*Total*	*Night*
100%	5/11	43	38	11	10	16	5
	3/11	43	38	12	10	17	5
	2/11	43	38	11	12	17	6
	1/11	43	38	11	10	17	5
80%	4/7	43	23	11	10	16	5
	3/7	43	23	12	10	17	5
60%	1/1	63	0	16	0	16	0
75% Night Nurses	1/3	0	96	0	32	16	16
	1/3	0	95	0	32	16	16
	1/3	0	93	0	32	16	16
75%	1/1	40	15	16	0	16	0
50%	3/4	40	15	16	0	16	0
	1/4	76	0	32	0	16	0

Fairness in the calendar-based schedule is accounted for in the same way as the cyclical schedule. Consequently, we get an equal distribution of E- and N-shifts during weekdays for nurses within the same group. Since the nurses work every third weekend, some nurses must work one more weekend due to the number of weeks in the planning horizon being 52. This results in a few differences of 1 regarding E-shifts and N-shifts during weekends, total number of weekends and total night weekends. However, because the rotation of every third weekend is continued after the planning horizon, this is expected to even out in next year’s schedule, such that nurses working 16 weekends in this plan will work 17 weekends in the next. The shifts regarding summer nurses and weekend nurses are allocated less evenly, as showed in table 7.4.

Table 7.4: *Distribution of unpopular shifts in Calendar-Based schedule for summer and weekend nurses*

		Weekdays		**Weekend**		**Work Weekends**	
Position	*Portion*	*E-shift*	*N-shift*	*E-shift*	*N-shift*	*Total*	*Night*
20% Weekend Nurses	1/6	23	3	16	2	17	1
	1/6	23	6	16	2	17	1
	1/6	23	4	16	2	17	1
	1/6	23	4	17	2	16	1
	1/6	23	6	17	2	16	1
	1/6	20	4	16	2	17	1
100% Summer Nurses	2/5	7	5	0	4	4	2
	1/5	4	6	1	4	4	2
	1/5	6	4	0	4	4	2
	1/5	5	7	0	4	4	2

The weekend nurses are assigned to work full time for a few weeks during the summer. In this period, the main concern is minimizing under-coverage of demand. Hence, the model did not account for fairness regarding shift distribution between the weekend nurses or summer nurses in this period. The unpopular shifts turned out to be distributed to a sufficient degree of fairness in the summer period, however there is potential for improvement. For example, one of the summer nurses received 7 N-shifts, while another received 4. A better allocation could be for example 5 or 6 N-shifts for each summer nurse. However, the number of work weekends and night weekends are distributed as evenly as possible. Furthermore, there is an even distribution of unpopular shifts the rest of the year for the weekend nurses.

With regard to the total results, we will argue that the calendar-based schedule model performs to a high degree with respect to fairness. There is potential to increase fairness during the summer period, however, this was not prioritized in this case.

7.4 Coverage of demand

In the cyclical schedule, coverage of demand is handled by hard constraints such that the minimum requirements of workforce are ensured. Over-coverage is accounted for by limiting the surplus of nurses on a shift to be maximum one more than demand. This ensures that over-coverage is evenly distributed. Furthermore, the model allocates an extra nurse to D-shifts instead of E- and N-shifts as the last ones are considered unpopular shifts. It would not be ideal to eliminate over-coverage completely, because it is needed for the nurses to reach a number of work hours closer to the contracted hours. Furthermore, it can be convenient to have a surplus of nurses in case of sickness. The graph below shows how workforce is scheduled compared to the manpower plan.

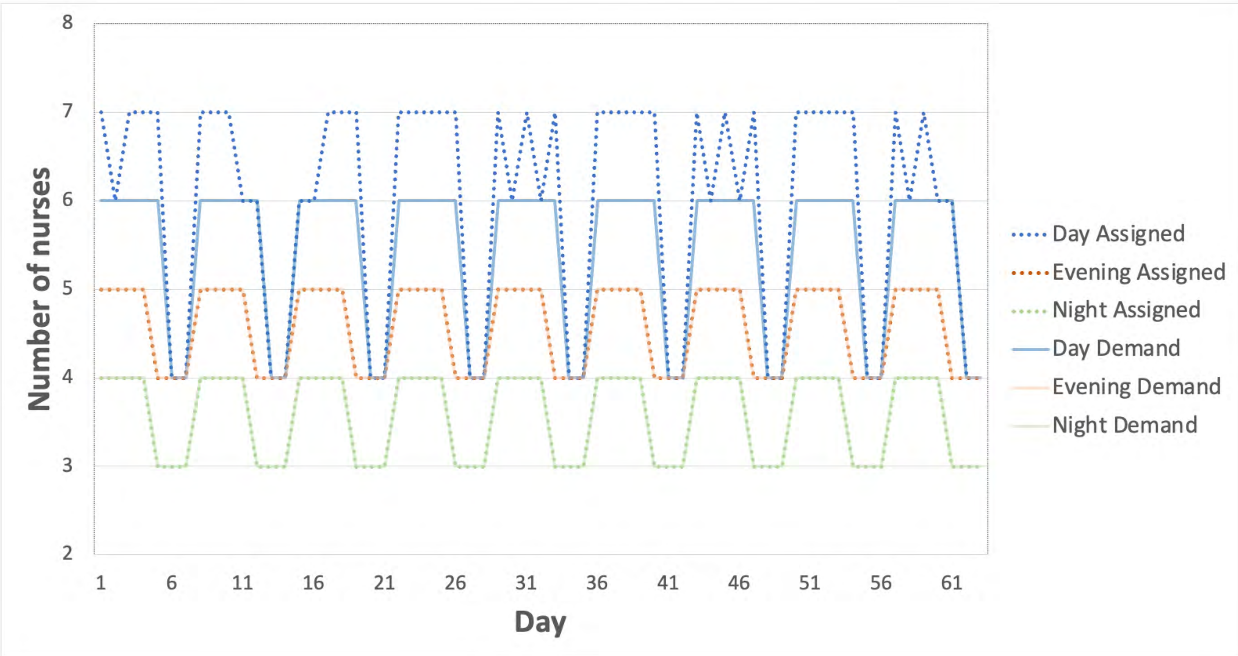

Figure 7.3: *Comparison of demand and number of nurses assigned each shift in Cyclical Schedule*

In the calendar-based schedule, coverage of demand is approached in the same manner as the cyclical schedule, apart from the summer period. Due to vacations, there is a shortage of available workforce. Although five temporary nurses are employed, the required workforce is not obtained. This is handled by allowing one less nurse to be assigned a work shift than what is originally required in the manpower plan. However, this can only happen in one shift during the day, and the number of days with a shortage is minimized by the model.

In summary, coverage of demand is guaranteed in the cyclical schedule, and outside of the summer period in the calendar-based schedule. Furthermore, under-coverage has been minimized to ensure it happens as rarely as possible. Consequently, we conclude that coverage of demand is obtained to a high degree.

8. Discussion

8.1 Limitations

In this master's thesis we have solved two nurse scheduling problems for a realistic case provided by the staffing department at Haukeland University Hospital. The models have proven to generate good quality schedules that could significantly reduce the time spent on creating a schedule. However, we need to take into considerations the fact that this case, although realistic, represents a simpler department structure. Consequently, the model we derived for this problem does not consider certain nuances that differs between departments, even at the same hospital. Some department structures are far mor complex than others, for example regarding skill levels. Although we implemented the difference in competency to some degree, many real-life departments involve a workforce with a significantly broader variety of skill levels. In turn, their services require a particular number of employees with a certain competence level which complicates the scheduling process even further.

The models have provided insight to the staff at Haukeland on how optimization and computational tools can be used to deal with the complex scheduling problem. However, if the models would be implemented as a decision support tool, ease of use is an important consideration. The models can be utilized on different departments by changing the data input or model algorithm. However, this requires a certain mathematical programming expertise, which the majority of head nurses who manually creates schedules today usually do not possess. Consequently, the models are not necessarily easy to implement as a decision support tool in practice at a hospital.

8.2 Future Work

During the course of working with this problem, a couple of interesting topics came to mind. In this chapter, we propose topics for future research derived from the work in this thesis.

The models designed in this thesis considers general preferences but does not account for personal request specifically. The nurses may have requests regarding for instance certain days off, preferred shifts patterns and when to have vacation. It would be interesting to explore further how an automated approach could facilitate for such personal requests. There is also a lot of potential to expand the problem to account for more complicated department structures, including factors such as skill levels or a higher number of employees.

Additionally, the optimization based two-stage heuristic suggested in this thesis is, to the best of our knowledge, somewhat unique. It could be interesting to see how this approach performs on a different case. Another topic could be to explore this approach further. Alternatively, one could treat the weekend and weekdays as two separate shift scheduling problems, such that the 1-stage model not only assigns work weekends, but also allocates the specific shifts. That way, the 2-stage model would still have to take some input from the 1-stage model, but it could completely disregard assigning shifts during the weekend. Furthermore, the objective of minimizing δ^+ and δ^- could be included in the 1-stage model instead, as these decision variables only regard the weekend shifts. These steps would lighten the workload for the stage 2 model, possibly reducing solving time or make room for other objectives.

Lastly, the models derived in this thesis requires some manual allocation of shifts to achieve the right number of hours before the schedules could be ready for practical implementation. Consequently, it would be interesting to incorporate different shift types of different lengths. For example, including some alternative short shifts could be very useful. This would allow for more flexibility in terms of calculating average work hours, thus decreasing the difference in contracted and scheduled hours. Incorporating such short shifts and making sure they are merely assigned to achieve the right number of hours, could potentially eliminate the need for manual adjustments in the final steps.

9. Conclusion

The purpose of this thesis was to solve a nurse scheduling problem by formulating a mathematical optimization model that can create realistic and satisfactory schedules in a way that is preferable to manual methods.

The models proposed in this thesis are formulated in line with the scheduling principles at Haukeland University Hospital, and different approaches were utilized to accommodate two forms of scheduling, cyclical and calendar based. For the cyclical schedule, we formulated an optimization model using mixed integer linear programming. This model was optimized directly using the solver of the software. The calendar-based schedule was solved by designing an optimization-based heuristic of decomposing the problem into two sub problems. The first problem involved allocating vacations and work weekends, as well as determining if the nurse work night shifts during the weekend or not. The second problem assigned nurses to specific shifts while using the results from the 1-stage model as fixed input parameters.

The results show that the schedules derived from the models perform well with respect to the criteria, which were computational time, fairness, coverage of demand and preferred practices. The most significant benefit of the decision models is the considerable reduction in computational time. The manual method generally requires 1-2 weeks to create a cyclical schedule, and 4-6 weeks for a calendar-based schedule. The cyclical schedule model obtained an optimal solution in less than 2 hours. The optimization-based heuristic approach for the calendar-based schedule obtained a good solution within 2 hours.

References

Akkermans, A., Post, G., & Uetz, M. (2021). Solving the shift and break design problem using integer linear programming. *Annals of Operations Research*, *302*(2), 341–362. https://doi.org/10.1007/s10479-019-03487-6

AMPL (2020). AMPL API. Retrieved from https://ampl.com/products/ampl/ide/

Arbeidsmiljøloven. (2005). Lov om arbeidsmiljø, arbeidstid og stillingsvern m.v. (LOV-2005-06-17-62). Lovdata. https://lovdata.no/dokument/NL/lov/2005-06-17-62

Beckmann, F. R., & Klyve, K. K. (2016). *Optimisation-Based Nurse Scheduling for Real-Life Instances*. Norwegian University of Science and Technology.

Berland, A. (2019). Årsplan - ingen quick-fix. *Norsk Sykepleierforbund*. Retrieved from: https://www.nsf.no/fylke/rogaland/nyheter/arsplan-ingen-quick-fix

Blytt, K. M., Bjorvatn, B., Moen, B. E., Pallesen, S., Harris, A., & Waage, S. (2022). The association between shift work disorder and turnover intention among nurses. *BMC Nursing*, *21*(1), 143. https://doi.org/10.1186/s12912-022-00928-9

Brucker, P., Qu, R., & Burke, E. (2011). Personnel scheduling: Models and complexity. *European Journal of Operational Research*, *210*(3), 467–473. https://doi.org/10.1016/J.EJOR.2010.11.017

Burke, E. K., de Causmaecker, P., Berghe, G. vanden, & van Landeghem, H. (2004). The State of the Art of Nurse Rostering. *Journal of Scheduling*, *7*(6), 441–499. https://doi.org/10.1023/B:JOSH.0000046076.75950.0b

Dantzig, G. B. (1954). Letter to the Editor—A Comment on Edie's "Traffic Delays at Toll Booths." *Journal of the Operations Research Society of America*, *2*(3), 339–341. https://doi.org/10.1287/opre.2.3.339

Dolonen, K. A., & Reppen, N. K. (2022). *Norge mangler 5900 sykepleiere og 700 spesialsykepleiere*. Retrieved from: https://sykepleien.no/2022/05/norge-mangler-5900-sykepleiere-og-700-spesialsykepleiere

Edie, L. C. (1954). Traffic Delays at Toll Booths. *Journal of the Operations Research Society of America*, *2*(2), 107–138. https://doi.org/10.1287/opre.2.2.107

Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, *153*(1), 3–27. https://doi.org/10.1016/S0377-2217(03)00095-X

Heyerdal, R. (2019). Søvnekspert advarer mot sein-tidlig-vakter: – Like belastende eller til og med verre enn nattskift. *FriFagbevegelse*. Retrieved from: https://frifagbevegelse.no/forside/sovnekspert-advarer-mot-seintidligvakter--like-belastende-eller-til-og-med-verre-enn-nattskift-6.158.626483.1bcc398835?fbclid=IwAR0Q4-ruXuVu81CRzhEDbo7d-Be5Yv3eyOgejNjTZs_lPHS9Tksk8Ihw9AM

Myklathun, K. H. (2022). *NAVs bedriftsundersøkelse 2022 - Stor mangel på arbeidskraft*. Retrieved from: https://www.nav.no/no/nav-og-samfunn/kunnskap/analyser-fra-nav/arbeid-og-velferd/%20arbeid-og-velferd/bedriftsundersokelsen?fbclid=IwAR24FdklOqMvsdi3Mm3Aw9pN8iLtpNnqINuyWTJcOzQOO3uFQ58XkW_DYPY

Ose, S. O., Færevik, H., Håpnes, T., & Øyum, L. (2022). Perceived Causes of Work-Related Sick Leave Among Hospital Nurses in Norway: A Prepandemic Study. *Safety and Health at Work*, *13*(3), 350–356. https://doi.org/https://doi.org/10.1016/j.shaw.2022.04.002

Stølevik, M., Nordlander, T., & Riise, A. (2011). *A mathematical model for the nurse rostering system*.

van den Bergh, J., Beliën, J., de Bruecker, P., Demeulemeester, E., & de Boeck, L. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research*, *226*(3), 367–385. https://doi.org/10.1016/J.EJOR.2012.11.029

Zanda, S., Zuddas, P., & Seatzu, C. (2018). Long term nurse scheduling via a decision support system based on linear integer programming: A case study at the University Hospital in Cagliari. *Computers & Industrial Engineering*, *126*, 337–347. https://doi.org/10.1016/J.CIE.2018.09.027

Appendices

Appendix 1: AMPL .mod-file for Cyclical Schedule Model

```
# SETS
# Sets of nurses
set N ordered; #Set of all nurses
set NN; #Set of nurses that only works night shifts
set NW; #Set of nurses that only works weekend
set NE; #Set of nurses that only works evening shifts
set NnoN; #Set of nurses who should not work night shifts
set NnoNW; #Set of nurses not working night shift weekends
set NmoreE; #Set of nurses who should have more evening shifts than day shifts on
weekdays
set Neq; #Set of nurses who should have equal amount of evening and night shifts
on weekdays
set N80plus; #Set of 100% and 80% nurses
set N100; #Set of 100% nurses
set N80; #Set of 80% nurses

# Sets of shifts
set S; #Set of all shift types
set Sw; #Set of work shifts
set So; #Set of free shift
set PI within {Sw,Sw}; #Set of incompatible shifts on consecutive days
set PU within {S,S,S,S}; #Set of unwanted shift patterns of consecutive days

# Sets of days of the week
set L; #Set of names for day of the week (monday,tuesday...Sunday)

# PARAMETERS
param QD; #Number of days in the planning period
param QW; #Number of weeks in the planning period
param QWKD; #Number of weekends every nurse should work during the planning period
param QNWKD; #Number of night shifts during weekend
param QHSM; #Number of hours from a Sunday night shift that occurs the following
Monday
param MH; #Max weekly hours
param MNE; #Max shifts in a row that are all night shifts or all evening shifts
param maxWS; #Max number of work shifts per nurse per week
param maxEN; #Max number og E- and N-shifts per nurse per week
param maxST; #Max number of shifts in each work shift type per nurse per week
param maxWN; #Max number of weekend nurses assigned to work the same weekend
param maxz; #Max deviation between contracted hours and scheduled hours

param A{Sw,L}; #Demand for shift s on day of the week l
param H{S}; #Hours per shift
param E{N}; #Goal work hours per week for nurse n
param Beta{S,S}; #If there is enough time between two shifts before and after a F1
day

# MORE SETS
# Sets of days and weeks
set D = 1..QD; #Set of days
set W = 1..QW; #Set of weeks
set WD{W}; #Set of days in weeks
set I; #Set of days in weekends
set II = D diff I; #Set of weekday days
set III; #Set of weekdays minus fridays
set DW{W}; #Set of days d in weekend w
set SW{W}; #Set of sundays in week W
set Day{L}; #Set of days in each day of the week
```

```
set Sunday; #Set of sundays

#VARIABLES
var x{N,S,D} binary; #1 if nurse n works shift s on day d
var y{N} ; #Actual average hour per week for nurse N
var z{N}>=0; #number of hours missing to give the nurse the correct amount of
hours
var h{N,W} binary; #If nurse n works weekend h
var hn{N,W} binary; #If nurse n works night weekend h
var shift{N,S,W} binary; #1 if nurse n works shift s at least once in week w
var qn{N}>= 0; #The number of night shifts for nurse n during weekdays
var qa{N}>= 0; #The number of evening shifts for nurse n during weekdays
var D1{N} >= 0; #If the difference in D- and E-shifts during weekend is >0 for
nurse n: D1>0, otherwise D1 = 0
var D2{N} <= 0; #If the difference in D- and E-shifts during weekend is <0 for
nurse n: D2<0, otherwise D2 = 0
var ED{N,D} binary; #If nurse n works A on day d and D on day d+1


# OBJECTIVE FUNCTION
minimize Objective:
  sum{n in NW} z[n] +
  sum{n in NN} z[n] +
  sum{n in N, d in D}ED[n,d]+
  sum{n in N}(D1[n]) -
  sum{n in N}(D2[n]);


# CONSTRAINTS
subject to

#------------------------------------ HOURS---------------------------------------

# BALANCE: CALCULATING AVERAGE HOURS WORKED PER WEEK

# For all nurses

       Balance1{n in N}:
       sum{s in S, d in D}(x[n,s,d]*H[s]) = y[n]*QW; #Average hours per week
assigned to nurse n


# DEVIATION: ACTUAL HOURS VS CONTRACTED HOURS

# For all nurses

       Actualhours{n in N}:
       E[n]*QW = z[n] + y[n]*QW;  #Deviation actual hours to contracted hours


# MAXIMUM DEVIATION
       maxdeviation{n in N: n not in NW and n not in NN}:
       z[n]<=maxz;


# MAX HOURS PER WEEK

       MaxWeeklyHours{n in N, w in W: w>1}:
       sum{s in S, d in WD[w]}(x[n,s,d]*H[s])-sum{d1 in
SW[w]}(x[n,"N",d1]*QHSM)+sum{d2 in SW[w-1]}(x[n,"N",d2]*QHSM)<=MH;

       MaxWeeklyHours2{n in N}: #Last week into first week
       sum{s in S, d in WD[1]}(x[n,s,d]*H[s])-sum{d1 in
SW[1]}(x[n,"N",d1]*QHSM)+sum{d2 in SW[QW]}(x[n,"N",d2]*QHSM)<=MH;
```

```
#--------------------------------COVERAGE CONSTRAINTS-----------------------------

# CAN ALLOW ONE MORE ON DAY SHIFTS

	DemandConstraint{l in L, d in Day[l]}:
	A["D",l] <= sum{n in N}x[n,"D",d] <= A["D",l]+1; #Number of nurses working
is equal to demand or one more

# NUMBER OF NURSES SHOULD EQUAL STAFFING PLAN ON EVENING AND NIGHT SHIFTS

	DemandConstraint2{l in L, s in Sw,d in Day[l]:s<>"D"}:
	sum{n in N}x[n,s,d] = A[s,l]; #Number of nurses working equals the demand



#----------------------------F1 SHIFT: WEEKLY REST PERIOD-------------------------


# F1 SHIFT

	35off{n in N, s in S,d in D: d>2}:
	x[n,s,d-2]+x[n,"F1",d-1]+sum{s2 in S}(x[n,s2,d]*Beta[s,s2])<=2;

	1F1perweek{n in N,w in W}: #Only one F1 day per week, other free days must
be F day
	sum{d in WD[w]}x[n,"F1",d]=1;

	F1onSun{n in N, w in W, d in SW[w]}: #If nurse has a free weekend, the F1
day should be on sunday that week
	h[n,w]+x[n,"F1",d]=1;

# TIME OFF BEFORE WORK WEEKEND

	F1beforeWKD{n in N, w in W, d in SW[w]}: #If nurse works weekend, the F1 day
should be on thursday
	sum{s in Sw} x[n,s,d]-x[n,"F1",d-3]-x[n,"F1",d-2]=0;


# TIME OFF AFTER WORK WEEKEND

	OffatferWKD{n in N, d in Sunday: d<QD}:
	sum{s in Sw}x[n,s,d]+sum{s in Sw}x[n,s,d+1]+sum{s in Sw}x[n,s,d+2]<=2;

	OffatferWKD2{n in N}: #Last week into first week
	sum{s in Sw}x[n,s,QD]+sum{s in Sw}x[n,s,1]+sum{s in Sw}x[n,s,2]<=2;


#---------------------------GENERAL SHIFT CONSTRAINTS-----------------------------

# 1 SHIFT EVERY DAY

	1shiftperday{n in N, d in D}:
	sum{s in S}x[n,s,d]=1;


# MAX 5 DAYS OF WORKING SHIFTS PER WEEK

	5perweek{n in N,w in W}:
	sum{s in Sw, d in WD[w]}x[n,s,d]<=maxWS;


# NO MORE THAN 4 EVENING/NIGHT SHIFTS PER NURSE PER WEEK

	NightEvening{n in N, w in W: n not in NN}:
	sum{d in WD[w]}x[n,"E",d]+sum{d in WD[w]}x[n,"N",d]<=maxEN;
```

```
# NO MORE THAN 3 SHIFTS OF EACH WORKING SHIFT TYPE PER NURSE PER WEEK

      Maxthree {n in N, s in Sw, w in W: n not in NN}:
      sum{d in WD[w]}x[n,s,d]<=maxST;


#---------------------------UNWANTED COMBINATIONS OF SHIFTS----------------------

# INCOMPATIBLE SHIFTS ON CONSECUTIVE DAYS

      IncompatiblePairsOnCunsecutiveDays{n in N, (s,s2) in PI, d in D: d<QD}:
      x[n,s,d]+x[n,s2,d+1]<=1;

      IncompatiblePairsOnCunsecutiveDays2{n in N, (s,s2) in PI}:
      x[n,s,QD]+x[n,s2,1]<=1;


# EVENING-DAY SHIFTS AND DAY-EVENING SHIFTS (MAX 1 PER WEEK)

      EveningDay1{n in N, d in III}:
      x[n,"E",d]+x[n,"D",d+1]<=ED[n,d]+1;

      EveningDay2{n in N, w in W}:
      sum{d in WD[w]}ED[n,d]<=1;


# "JOJO"-SHIFTS


      JOJO{n in N, (s1,s2,s3,s4) in PU, d in D: d<QD-2}:
      x[n,s1,d]+x[n,s2,d+1]+x[n,s3,d+2]+x[n,s4,d+3]<=3;

      JOJO2{n in N, (s1,s2,s3,s4) in PU}:
      x[n,s1,QD-2]+x[n,s2,QD-1]+x[n,s3,QD]+x[n,s4,1]<=3;

      JOJO3{n in N, (s1,s2,s3,s4) in PU}:
      x[n,s1,QD-1]+x[n,s2,QD]+x[n,s3,1]+x[n,s4,2]<=3;

      JOJO4{n in N, (s1,s2,s3,s4) in PU}:
      x[n,s1,QD]+x[n,s2,1]+x[n,s3,2]+x[n,s4,3]<=3;


# MAX 2 N-SHIFTS ON WEEKDAYS FOR EVERYONE EXCEPT NN

      Max2inarow{n in N, d in D:d<QD-1 and d in III and n not in NN}:
      sum{a in 0..2}x[n,"N",d+a]<=2;

# MAX 3 CONSECUTIVE EVENING OR NIGHT SHIFTS

      Max3inarow{n in N, s in Sw, d in D:d<QD-2 and s <> "D" and n not in NN}:
      sum{a in 0..3}x[n,s,d+a]<=MNE;

      Max3inarow2{n in N, s in Sw: s<>"D" and n not in NN}:
      x[n,s,QD-2]+x[n,s,QD-1]+x[n,s,QD]+x[n,s,1]<=MNE;

      Max3inarow3{n in N, s in Sw: s<>"D" and n not in NN}:
      x[n,s,QD-1]+x[n,s,QD]+x[n,s,1]+x[n,s,2]<=MNE;

      Max3inarow4{n in N, s in Sw: s<>"D" and n not in NN}:
      x[n,s,QD]+x[n,s,1]+x[n,s,2]+x[n,s,3]<=MNE;

# AVOID WEEKS CONSISTING ONLY OF EVENING AND NIGHT SHIFTS

      Ifshift{n in N,s in Sw, w in W}:
      shift[n,s,w]*1000>=sum{d in WD[w]}x[n,s,d];
```

```
        Ifshift2{n in N,s in Sw,w in W}:
        shift[n,s,w]<=sum{d in WD[w]}x[n,s,d];

        Ifshift3{n in N, w in W: n not in NN and n not in NW}:
        shift[n,"D",w]+1>=shift[n,"E",w]+shift[n,"N",w];


#--------------------------------NURSE SPECIFIC CONSTRAINTS----------------------

# WEEKEND NURSES

        OnlyWkd{n in NW}:
        sum{s in Sw, d in III}x[n,s,d]+sum{d in Day["Friday"]}x[n,"D",d]=0;

# NIGHT SHIFT NURSES

        OnlyN{n in NN, s in Sw, d in D: s<>"N"}:
        x[n,s,d]=0;


# GROUP OF NURSES THAT SHOULD NOT HAVE NIGHT SHIFTS DURING WEEKEND

        NoNWkd{n in NnoNW, d in I}:
        x[n,"N",d]=0;


# GROUP OF NURSES THAT SHOULD NOT HAVE NIGHT SHIFTS AT ALL

        NoNight{n in NnoN, d in D}:
        x[n,"N",d]=0;


# GROUP OF NURSES THAT SHOULD ONLY HAVE EVENING SHIFTS

        OnlyEvening:
        sum{n in NE, s in Sw, d in D: s<>"E"}x[n,s,d]=0;


 # GROUP OF NURSES THAT SHOULD HAVE MORE EVENING SHIFTS THAN DAY SHIFTS DURING
WEEKDAYS

        MoreEthanN{n in NmoreE}:
        sum{d in II}x[n,"D",d]+1<=sum{d in II}x[n,"E",d];


#-------------------------------NIGHT SHIFTS CONSTRAINTS-------------------------

# NIGHT SHIFT FOLLOWED BY NIGHT SHIFT OR DAY OFF

        AfterNshift1{n in N, d in D: d<QD-1}: #A night shift should always be
followed by another night shift or a sleep day + day off
        x[n,"N",d] <= x[n,"N",d+1]+(1-sum{s in Sw}x[n,s,d+2]);

        AfterNshift2{n in N}: #Last week into first week
        x[n,"N",QD-1] <= x[n,"N",QD]+(1-sum{s in Sw}x[n,s,1]);

        AfterNshift3{n in N}: #Last week into first week
        x[n,"N",QD] <= x[n,"N",1]+(1-sum{s in Sw}x[n,s,2]);


# AVOID DAY OFF-NIGHT SHIFT-DAY OFF

        OffNOff1{n in NN, d in D: d<QD-1}:
        sum{s in So}x[n,s,d]+x[n,"N",d+1]+sum{s in So}x[n,s,d+2]<=2; #Cannot have
day off - N - day off
```

```
	OffNOff2{n in NN}: #Last week into first week
	sum{s in So}x[n,s,QD]+x[n,"N",1]+sum{s in So}x[n,s,2]<=2;  #Cannot have day
off - N - day off


#------------------------------WEEKEND CONSTRAINTS------------------------------

# IF WORK WEEKEND THEN NURSE MUST WORK SATURDAY AND SUNDAY

	Weekend1{n in N, w in W, d in DW[w]}:
	h[n,w]=sum{s in Sw}x[n,s,d];

	Weekend2{n in N, w in W:w<QW-2}:
	h[n,w]=h[n,w+3];

	Weekend3{n in N}:
	sum{w in W}h[n,w]=QWKD;


# IF WORK EVENING OR NIGHT FRIDAY THEN NURSE MUST WORK THAT WEEKEND

	EvNiFri{n in N, s in Sw, d in Sunday: s<>"D"}:
	sum{ss in Sw}x[n,ss,d]>=x[n,s,d-2];


# IF NURSE WORKS NIGHT SHIFT WEEKEND THEN NURSE HAS NIGHT SHIFT FRIDAY, SATURDAY
AND SUNDAY

	WeekendN{n in N80plus}:
	sum{w in W, d in DW[w]}x[n,"N",d] = QNWKD;

	WeekendN2{n in N, w in W, d in SW[w]}:
	hn[n,w]*3 = x[n,"N",d]+x[n,"N",d-1]+x[n,"N",d-2];


# CANNOT HAVE SAME SHIFT FRIDAY,SATURDAY AND SUNDAY EXCEPT WHEN NIGHT SHIFT
	Weekend4{n in N,s in Sw, d in Sunday:s <>"N"}:
	x[n,s,d]+x[n,s,d-1]+x[n,s,d-2]<=2;


 # MAXIMUM TWO WEEKEND NURSES WORKING THE SAME WEEKEND

	Max2sameWKD{w in W}:
	sum{n in NW}h[n,w]<=maxWN;


# WEEKEND NURSES SHOULD NOT WORK SAME SHIFT AS ANOTHER WEEKEND NURSE

	Notsameshift{n in NW, nn in NW, d in D,s in Sw: n<>nn}:
	x[n,s,d]+x[nn,s,d]<=1;


#ONE NURSE SHOULD NOT WORK EVENING BOTH FRIDAY AND SATURDAY OF SAME WEEKEND

	FriSat{n in N, d in Day["Friday"]}:
	x[n,"E",d]+x[n,"E",d+1]<=1;


#-----------------------------FAIR DISTRIBUTION OF SHIFTS-------------------------

# NIGHT SHIFTS DURING WEEKDAYS (MON-FRI)

	NumNightShifts{n in N}:
	sum{d in II}x[n,"N",d]=qn[n];
```

```
	NumNightShifts2{n in N100, nn in N100: n<>nn}:
	qn[n]=qn[nn];

	NumNightShifts3{n in N80, nn in N80: n<>nn}:
	qn[n]=qn[nn];

	NumNightShifts4{n in Neq, nn in Neq: n<>nn}:
	qn[n]=qn[nn];

	NumNightShifts5{n in N100, nn in N80}:
	qn[n]>=qn[nn]; #N100 nurses must work more night shifts than N80 nurses

	NumNightShifts6{n in N80, nn in Neq}:
	qn[n]>=qn[nn]+1; #N80 nurses must work more night shifts than Neq nurses

# EVENING SHIFTS DURING WEEKDAYS (MON-FRI)

	NumEveningShifts{n in N}:
	sum{d in II}x[n,"E",d]=qa[n];

	NumEveningShifts2{n in N100, nn in N100: n<>nn}:
	qa[n]=qa[nn];

	NumEveningShifts3{n in N80, nn in N80: n<>nn}:
	qa[n]=qa[nn];

	NumEveningShifts4{n in Neq, nn in Neq: n<>nn}:
	qa[n]=qa[nn];

	NumEveningShifts5{n in N100, nn in N80}:
	qa[n]>=qa[nn];  #N100 nurses must work more evening shifts than N80 nurses

	NumEveningShifts6{n in N80, nn in Neq}:
	qa[n]>=qa[nn]+1;  #N80 nurses must work more evening shifts than Neq nurses


# EQUAL DISTRIBUTION OF DAY AND EVENING SHIFTS DURING WEEKENDS

	D1D2{n in N:n <> "A24"}:
	D1[n]+D2[n]=sum{d in I}x[n,"D",d]-sum{d in I}x[n,"E",d];
```

Appendix 2: AMPL .**mod-file** for Stage 1 Calendar-Based Schedule

```
# SETS
# Sets of nurses
set N ordered; #Set of nurses
set NN; #Set of nurses that only works night shifts
set NW; #Set of nurses that only works weekend
set NS; #Set of summer nurses
set NnoNW; #Set of nurses not working night weekends at all
set N80plus; #Set of 100% and 80% nurses

# Sets of shifts
set SC; #Set of shift category, night or day/evening

# Sets of weeks
set Wwinter; #Set of winter vacation weeks
set Wsummer; #Set of summer vacation weeks
set WSsummer; #Set of summer start vacation weeks
set Wautumn; #Set of autumn vacation weeks
set WSchool; #Set of school vacation weeks during winter and autumn

# PARAMETERS
param QW; #Number of weeks
param A{SC}; #Demand of shift s
param FS; #Number of nurses in each summer vacation period
param minEL{SC}; #Number of skilled nurses required in shift category s
param maxWN; #Max number of weekend nurses assigned to work the same weekend
param QNWKD; #Number of night shifts during weekend
param QvacSchool{WSchool}; #Number of nurses who should have winter/autumn
vacation during school vacation

# MORE SETS
set W = 1..QW; #Set of weekends
set SN dimen 2;

#VARIABLES
var x{N,SC,W} binary;
var h{N,W} binary; #1 if nurse works weekend w
var hn{N,W} binary; #1 if nurse works night weekend w
var nv{N,W} binary; #summer vacation


# OBJECTIVE FUNCTION
minimize Objective:
      sum{w in Wsummer, c in SC}(A[c]-sum{n in N}x[n,c,w]);


subject to
#---------------------------------- GENERAL CONSTRAINTS --------------------------

      MaxOneShift{n in N, w in W}:
      sum{c in SC}x[n,c,w]<=1;

      IndicatorVariable1{n in N, w in W}: #1 if nurse n works weekend w
      h[n,w]=sum{c in SC}x[n,c,w];

      IndicatorVariable2{n in N, w in W}:  #1 if nurse n works nights weekend w
      hn[n,w]=x[n,"N",w];


#--------------------------------COVERAGE CONSTRAINTS----------------------------
```

```
# DEMAND CONSTRAINTS NORMAL WEEKENDS

        DemandConstraint{c in SC, w in W: w not in Wsummer}:
        sum{n in N}x[n,c,w] = A[c]; #Number of nurses working equals the demand


# DEMAND CONSTRAINTS SUMMER WEEKENDS

        DemandConstraint2{w in Wsummer}:
        A["DE"]-1 <= sum{n in N}x[n,"DE",w] <= A["DE"];

        DemandConstraint3{w in Wsummer}:
        sum{n in N}x[n,"N",w] = A["N"];


# ALWAYS AT LEAST 1 EXPERIENCED NURSE ON A SHIFT

        DemandConstraint4{w in W, c in SC}:
        sum{n in N: n not in NS and n not in NW}x[n,c,w] >= minEL[c];


#----------------------------DISTRIBUTION OF WORK WEEKENDS-------------------------


# TWO WEEKENDS BETWEEN EACH WORK WEEKEND

        Weekend{n in N, w in W: w<QW-1 and n not in NS}:
        sum{a in 0..2}h[n,w+a]<=1;

        Weekend2{n in N, w in W: w<QW-8 and n not in NS and n not in NN and n not in
NW}:
        sum{a in 0..8}hn[n,w+a]<=1;


# SUMMER EMPLOYEES

        Weekend3{n in NS, w in Wsummer}:
        h[n,w]+h[n,w+1]<=1;

        Weekend4{n in NS, w in Wsummer}:
        sum{a in 0..2}hn[n,w+a]<=1;

        Weekend5{n in NS, w in W: w not in Wsummer}:
        h[n,w]=0;

        SummerNurses{n in NS, (d1,d2) in SN}: #Must work consecutive weeks of
maximum 7 weeks
        h[n,d1]+h[n,d2]<=1;


# WEEKEND NURSES DURING SUMMER

        Weekend6{n in NW, w in Wsummer}:
        sum{a in 0..8}hn[n,w+a]<=1;

        Weekend7{n in NW}:
        sum{w in W:w not in Wsummer}hn[n,w]=0;


# NIGHT SHIFT NURSES AND NURSES WHO SHOULD NOT HAVE NIGHT SHIFTS

        NightWorkers{n in NN, w in W}:
        x[n,"DE",w]=0;

        NotnightW{n in NnoNW}:
```

```
      sum{w in W}hn[n,w]=0;


# MAXIMUM TWO WEEKEND EMPLOYEES WORKING THE SAME WEEKEND

      Max2sameWKD{w in W:w not in Wsummer}:
      sum{n in NW}x[n,"DE",w]<=2;


#----------------------------------- VACATION ------------------------------------

# NO SHIFT IF VACATION

      Vacation{n in N, w in W: n not in NS}:
      nv[n,w]+h[n,w]<=1;

#WINTER AND AUTUMN VACATION

      winter{n in N:n not in NS and n not in NW}:
      sum{w in Wwinter}nv[n,w]=1;

      autumn{n in N:n not in NS and n not in NW}:
      sum{w in Wautumn}nv[n,w]=1;


# MOST PEOPLE SHOULD HAVE VACATION DURING SCHOOL VACATION WEEKS

      winterautumn{w in WSchool}:
      sum{n in N: n not in NS and n not in NW}nv[n,w]= QvacSchool[w];


# EQUAL DISTRIBUTION OF NURSES WITH VACATION THE REMAINING VACATION WEEKS

      winter3{w in Wwinter: w<>52}:
      1<=sum{n in N: n not in NS and n not in NW}nv[n,w]<=2;

      autumn3{w in Wautumn: w<>32}:
      1<=sum{n in N: n not in NS and n not in NW}nv[n,w]<=2;


# EQUAL DISTRIBUTION OF NURSES WITH SUMMER VACATION DURING SUMMER WEEKS

      Summer{w in WSsummer}:
      sum{n in N: n not in NS and n not in NW}nv[n,w]=FS;

      Summer2{n in N, w in WSsummer:n not in NS and n not in NW}:
      nv[n,w]*2=nv[n,w+1]+nv[n,w+2];

      Summer3{n in N: n not in NS and n not in NW}:
      sum{w in WSsummer}nv[n,w]=1;



#-------------------------------FAIRNESS CONSTRAINTS -----------------------------

# EQUAL DISTRIBUTIN OF NIGHT SHIFT WEEKENDS

      Min5each{n in N80plus}:
      sum{w in W}hn[n,w]>=QNWKD;
```

Appendix 3: AMPL **.mod-file** for Stage 2 Calendar-Based Schedule

```
# SETS
# Sets of nurses
set N ordered; #Set of nurses
set NW; #Set of nurses that only works weekend
set NN; #Set of nurses that only works night shifts
set NE; #Set of nurses that only works evening shifts
set NnoN;  #Set of nurses who should not work night shifts
set NmoreE; #Set of nurses not working night shift weekends
set NS; #Set of summer nurses
set N80plus; #Set of 100% and 80% nurses
set Neq; #Set of nurses who should have equal amount of evening and night shifts
on weekdays
set N100; #Set of 100% nurses
set N80; #Set of 80% nurses

# Sets of shifts
set S; #Set of shift types
set Sw; #Set of work shifts
set So; #Set of free shift
set PI within {Sw,Sw}; #Set of incompatible shifts on consecutive days
set PU within {S,S,S,S}; #Set of unwanted shift patterns of consecutive days
set SN dimen 2; #
set WN dimen 2; #

# Sets of days and weeks
set L; #Set of weekday names, monday,tuesday...
set Day{L}; #Set of days d in D that is a monday,tuesday...

# PARAMETERS
param A{Sw,L}; #Demand for shift S on day d
param H{S}; #Hours per shift
param E{N}; #Average hours per week for nurse n
param Beta{S,S}; # If there is enough time between two shifts before and after a
F1 day
param QD; #Number of days in the planning period
param QW; #Number of weeks in the planning period
param QWS; #Number of work weeks for summer nurses
param QWW; #Number of full time work weeks during summer for weekend nurses
param QVW; #Total number of vacation weeks to be assigned to each nurse
param MH; #Max weekly hours
param MNE; #Max night shifts in a row
param maxWS; #Max number of work shifts per nurse per week
param maxEN; #Max number og E- and N-shifts per nurse per week
param maxST; #Max number of shifts in each work shift type per nurse per week
param maxz; #Max deviation between contracted hours and scheduled hours
param QHSM; #Number of hours from a Sunday night shift that occurs the following
Monday
param Perc{N}; #Percentage of fulltime position for nurse n

# MORE SETS
set HD ordered;
set SummerD;
set HDS := HD union SummerD;
set bfHD; #Day before a holiday
set D = 1..QD; #Set of days
set W = 1..QW; #Set of weeks
set Wsummer; #Set of summer weeks
set WSsummer; #Set of start weeks summer
set Wwinter; #Set of winter vacation weeks
set Wautumn; #Set of autumn vacation weeks
set WD{W}; #Set of days in weeks
set I; #Set of days in weekends
set II = D diff I; #Set of weekday days
```

```
set III; #Set of weekdays minus fridays
set DW{W}; #Set of days d in weekend w
set SW{W}; #Set of sundays in week W
set Sunday; #Set of sundays

param h{N,W}; #If nurse n works weekend w
param hn{N,W}; #If nurse n works night weekend w
param nv{N,W}; #If nurse has summer vacation week w

# VARIABLES
var x{N,S,D} binary; #1 if nurse n works shift s on day d
var y{N} ; #Actual average hour per week for nurse N
var z{N}>=0; #number of hours missing to give the nurse the correct amount of
hours
var qn{N}>= 0; #The number of night shifts for nurse n during weekdays
var qa{N}>= 0; #The number of evening shifts for nurse n during weekdays
var D1{N} >= 0; #If the difference in D- and E-shifts during weekend is >0 for
nurse n: D1>0, otherwise D1 = 0
var D2{N} <= 0;  #If the difference in D- and E-shifts during weekend is <0 for
nurse n: D2<0, otherwise D2 = 0
var ED{N,D} binary; #If nurse n works A on day d and D on day d+1
var F3{N,HD} binary; #1 if nurse has HD off
var F3w{N,HD} binary; #1 if nurse works day d in HD
var lack{S,D} binary; #1 if there is a lack of one nurse on shift s on day d
var hundred{NW,W} binary; #1 if nurse n has 100% work week
var sommer{NS,Wsummer} binary; #1 if nurse n works summer week w
var shift{N,Sw,W} binary; #1 if nurse n works shift s at least once in week w

# OBJECTIVE FUNCTION
minimize Objective:
  sum{n in N, d in D}ED[n,d]+
  sum{s in Sw, d in SummerD} lack[s,d] +
  sum{n in N}(D1[n]) +
  sum{n in N}(D2[n])*(-1);


# CONSTRAINTS
subject to
#------------------------------------ HOURS----------------------------------------

# BALANCE: CALCULATING AVERAGE HOURS WORKED PER WEEK

# For all nurses who are not weekend-nurses or summer-nurses
       Balance1{n in N:n not in NW and n not in NS}:
       sum{s in Sw, d in D}(x[n,s,d]*H[s])+(7.5*Perc[n]*sum{d in
HD}F3[n,d])+(35.5*Perc[n]*QVW) = y[n]*QW; #Average hours per week assigned to
nurse n

# For all weekend-nurses
       Balance2{n in NW}:
       sum{s in Sw, d in D}(x[n,s,d]*H[s])  = y[n]*QW;

# For all summer-nurses
       Balance3{n in NS}:
       sum{s in Sw, d in SummerD}(x[n,s,d]*H[s]) = y[n]*QWS;


# DEVIATION: ACTUAL HOURS VS CONTRACTED HOURS

# For all nurses who are not weekend-nurses or summer-nurses
       Actualhours{n in N:n not in NW and n not in NS}:
       E[n]*QW = z[n] + y[n]*QW;

# For all weekend-nurses
       Actualhours2{n in NW}:
       E[n]*(QW-QWW)+E["A1"]*QWW = z[n] + y[n]*QW;
```

```
# For all summer-nurses
	Actualhours3{n in NS}:
	E[n]*QWS = z[n] + y[n]*QWS;


# MAX HOURS PER WEEK

	MaxWeeklyHours{n in N, w in W: w>1}:
	sum{s in S, d in WD[w]}(x[n,s,d]*H[s])-sum{d1 in
SW[w]}(x[n,"N",d1]*QHSM)+sum{d2 in SW[w-1]}(x[n,"N",d2]*QHSM)<=MH;

	MaxWeeklyHours2{n in N}: #First week
	sum{s in S, d in WD[1]}(x[n,s,d]*H[s])-sum{d1 in
SW[1]}(x[n,"N",d1]*QHSM)<=MH;

# MAXIMUM DEVIATION
	maxdeviation{n in N}:
	z[n]<=maxz;

#-------------------------------COVERAGE CONSTRAINTS-----------------------------

# DEMAND CONSTRAINTS NORMAL DAYS

	DemandConstraint1{l in L, d in Day[l]:d not in HDS}:
	A["D",l] <= sum{n in N}x[n,"D",d] <= A["D",l]+1;

	DemandConstraint2{l in L, d in Day[l]:d not in HDS}:
	sum{n in N}x[n,"E",d] = A["E",l]; #Number of nurses working equals the
demand

	DemandConstraint3{l in L, d in Day[l]:d not in HDS and d not in bfHD}:
	sum{n in N}x[n,"N",d] = A["N",l]; #Number of nurses working equals the
demand


# DEMAND CONSTRAINTS SUMMER
	DemandConstraint4{l in L, d in Day[l]:d in SummerD}:
	A["D",l]-1 <= sum{n in N}x[n,"D",d] <= A["D",l]+1;

	DemandSummer{s in Sw, l in L, d in Day[l]:d in SummerD and s<>"D"}:
	A[s,l]-1 <= sum{n in N}x[n,s,d] <= A[s,l];

	Lack{d in SummerD}:
	sum{s in Sw}lack[s,d]<=1;

# Lack per shift per week

	Lack2{l in L, d in Day[l], s in Sw:d in SummerD}:
	lack[s,d]=A[s,l]-sum{n in N}x[n,s,d];

# DEMAND CONSTRAINTS HOLIDAYS

# Demand holidays is equal to demand on weekends
	DemandHD {s in Sw, d in HD}:
	sum{n in N}x[n,s,d] = A[s,"Sunday"];

# Demand the nightshift going into a holiday is equal to demand on night shifts
weekends
	DemandbfHD {d in bfHD}:
	sum{n in N}x[n,"N",d] = A["N","Sunday"];


#-------------------------------F1 SHIFT: WEEKLY REST PERIOD---------------------

# F1 SHIFT
```

```
	35off{n in N, s in S,d in D: d>2}:
	x[n,s,d-2]+x[n,"F1",d-1]+sum{s2 in S}(x[n,s2,d]*Beta[s,s2])<=2;

	1F1perweek{n in N,w in W}: #Only one F1 day per week, other free days must
be F day
	sum{d in WD[w]}x[n,"F1",d]=1;

	F1onSun{n in N, w in W, d in SW[w]}: #If nurse has a free weekend, the F1
day should be on sunday that week
	h[n,w]+x[n,"F1",d]=1;


# TIME OFF AFTER WORK WEEKEND

	OffatferWKD{n in N, d in Sunday: d<QD}: #If work sunday then monday or
tuesday off
	sum{s in Sw}x[n,s,d]+sum{s in Sw}x[n,s,d+1]+sum{s in Sw}x[n,s,d+2]<=2;


# TIME OFF BEFORE WORK WEEKEND

	F1beforeWKD{n in N, w in W, d in SW[w]}: #If nurse works weekend, the F1 day
should be on thursday or friday
	sum{s in Sw} x[n,s,d]-x[n,"F1",d-3]-x[n,"F1",d-2]=0;


#----------------------------GENERAL SHIFT CONSTRAINTS----------------------------

# 1 SHIFT EVERY DAY

	1shiftperday{n in N, d in D}:
	sum{s in S}x[n,s,d]=1;

# MAX 5 DAYS OF WORKING SHIFTS PER WEEK

	5perweek{n in N, w in W}:
	sum{d in WD[w], s in Sw}x[n,s,d]<=maxWS;

# NO MORE THAN 4 EVENING/NIGHT SHIFTS PER NURSE PER WEEK

	NightEvening{n in N, w in W: n not in NN}:
	sum{d in WD[w]}x[n,"E",d]+sum{d in WD[w]}x[n,"N",d]<=maxEN;


# NO MORE THAN 3 SHIFTS OF EACH WORKING SHIFT TYPE PER NURSE PER WEEK

	Maxthree {n in N, s in Sw, w in W: n not in NN}:
	sum{d in WD[w]}x[n,s,d]<=maxST;



#-------------------------------UNWANTED COMBINATIONS OF SHIFTS-------------------

# INCOMPATIBLE SHIFTS ON CONSECUTIVE DAYS

	IncompatiblePairsOnCunsecutiveDays{n in N, (s,s2) in PI, d in D: d<QD}:
	x[n,s,d]+x[n,s2,d+1]<=1;


# EVENING-DAY SHIFTS AND DAY-EVENING SHIFTS (MAX 1 PER WEEK)

	EveningDay1{n in N, d in III}:
	x[n,"E",d]+x[n,"D",d+1]<=ED[n,d]+1;

	EveningDay2{n in N, w in W}:
```

```
        sum{d in WD[w]}ED[n,d]<=1;


# "JOJO"-SHIFTS

        JOJO{n in N, (s1,s2,s3,s4) in PU, d in D: d<QD-2}:
        x[n,s1,d]+x[n,s2,d+1]+x[n,s3,d+2]+x[n,s4,d+3]<=3;


# CONSECUTIVE EVENING AND NIGHT SHIFTS

        Max2inarow{n in N, d in D:d<QD-1 and d in III and n not in NN}:
        sum{a in 0..2}x[n,"N",d+a]<=2;

        Max3inarow{n in N, s in Sw, d in D:d<QD-2 and s <> "D" and n not in NN}:
        sum{a in 0..3}x[n,s,d+a]<=MNE;



# AVOID WEEKS CONSISTING ONLY OF EVENING AND NIGHT SHIFTS

        Ifshift{n in N,s in Sw, w in W}:
        shift[n,s,w]*1000>=sum{d in WD[w]}x[n,s,d];

        Ifshift2{n in N,s in Sw,w in W}:
        shift[n,s,w]<=sum{d in WD[w]}x[n,s,d];

        Ifshift3{n in N, w in W: n not in NN and n not in NS and w not in Wsummer
and w not in Wwinter and w not in Wautumn}:
        shift[n,"D",w]+1>=shift[n,"E",w]+shift[n,"N",w];


#----------------------------INDIVIDUAL CONSTRAINTS-------------------------------

# NIGHT SHIFT NURSES

        OnlyN{n in NN, s in Sw, d in D: s<>"N"}:
        x[n,s,d]=0;


# SINGLE NURSE CONSTRAINTS

        NoNight{n in NnoN, d in D}:
        x[n,"N",d]=0;

        OnlyEvening:
        sum{n in NE, s in Sw, d in D: s<>"E"}x[n,s,d]=0;

        MoreEthanN{n in NmoreE}:
        sum{d in II}x[n,"D",d]+1<=sum{d in II}x[n,"E",d];


#------------------------------NIGHT SHIFTS CONSTRAINTS----------------------------

# NIGHT SHIFT FOLLOWED BY NIGHT SHIFT OR DAY OFF
        AfterNightShift{n in N, d in D: d<QD-1}:
        x[n,"N",d] <= x[n,"N",d+1]+(1-sum{s in Sw}x[n,s,d+2]);


# AVOID DAY OFF-NIGHT SHIFT-DAY OFF

        OffNOff{n in NN, d in D: d<QD-1}:
        sum{s in So}x[n,s,d]+x[n,"N",d+1]+sum{s in So}x[n,s,d+2]<=2; #Kan ikke ha
fri - natt - fri
```

```
#-------------------------------WEEKEND CONSTRAINTS---------------------------------

# IF WORK WEEKEND THEN NURSE MUST WORK SATURDAY AND SUNDAY

      Weekend{n in N, w in W, d in DW[w]}:
      h[n,w]=sum{s in Sw}x[n,s,d];

# IF WORK EVENING OR NIGHT FRIDAY THEN NURSE MUST WORK THAT WEEKEND

      EvNiFri{n in N, d in Sunday}: #If nurse n works afternoon or night shift
friday, they must work following weekend
      sum{s in Sw}x[n,s,d]>=sum{s in Sw:s<>"D"}x[n,s,d-2];


# IF NURSE WORKS NIGHT SHIFT WEEKEND THEN NURSE HAS NIGHT SHIFT FRIDAY, SATURDAY
AND SUNDAY

      WeekendNight{n in N, w in W, d in SW[w]}:
      hn[n,w]*3 = x[n,"N",d]+x[n,"N",d-1]+x[n,"N",d-2];

# CANNOT HAVE SAME SHIFT FRIDAY,SATURDAY AND SUNDAY EXCEPT WHEN NIGHT SHIFT
      Weekend2{n in N,s in Sw, d in Sunday:s <>"N"}:
      x[n,s,d]+x[n,s,d-1]+x[n,s,d-2]<=2;


# WEEKEND NURSES SHOULD NOT WORK SAME SHIFT AS ANOTHER WEEKEND NURSE

      Notsameshift{n in NW, nn in NW, s in Sw, d in I:d not in SummerD and n<>nn}:
      x[n,s,d]+x[nn,s,d]<=1;


#ONE NURSE SHOULD NOT WORK EVENING BOTH FRIDAY AND SATURDAY OF SAME WEEKEND

      FriSat{n in N, d in Day["Friday"]: n not in NS and n not in NW}:
      x[n,"E",d]+x[n,"E",d+1]<=1;

#-----------------------------------------VACATION------------------------------------

# NURSES CANNOT WORK WHEN THEY ARE ON VACATION
      Vacation{n in N,s in Sw, w in W, d in WD[w]: n not in NS and n not in NW}:
      x[n,s,d]<= 1-nv[n,w];

# SUMMER

# SUMMER EMPLOYEES ONLY WORK DURING SUMMER WEEKS

      Summerempl{n in NS, d in D:d not in SummerD}:
      sum{s in Sw}x[n,s,d]=0;


# EVERY WEEKEND NURSE HAS 6 WEEKS OF FULL-TIME WORK DURING THE SUMMER
      Notfulltimenormally{n in NW}:
      sum{w in W:w not in Wsummer}hundred[n,w] = 0;

      Fulltime{n in NW}:
      sum{w in Wsummer}hundred[n,w] = QWW;


# THE SIX FULL-TIME WEEKS SHOULD BE CONSECUTIVE

      NWfulltime{n in NW, (w1,w2) in WN}: #Must work consecutive weeks of maximum
6 weeks
      hundred[n,w1]+hundred[n,w2]<=1;
```

```
# WEEKEND EMPLOYEES SHOULD NOT BE ALLOWED TO WORK MON-THU AND FRIDAY DAY SHIFTS
UNLESS IT IS DURING THEIR FULL-TIME SUMMER WEEKS

	NWnormally{n in NW, w in W}:
	hundred[n,w]*1000>=sum{s in Sw, d in WD[w]:d in III}x[n,s,d]+sum{d in
WD[w]:d in Day["Friday"]}x[n,"D",d]+sum{d in SW[w]}(x[n,"N",d-2]+x[n,"N",d-
1]+x[n,"N",d]);


# SUMMER EMPLOYEES ONLY WORK 7 CONSECUTIVE WEEKS EACH DURING THE SUMMER

	Summerempl2{n in NS}:
	sum{w in Wsummer}sommer[n,w] = QWS;

	Summerempl3{n in NS, w in Wsummer}:
	sommer[n,w]*1000>=sum{s in Sw, d in WD[w]}x[n,s,d];

	Summerempl4{n in NS, (w1,w2) in SN}: #Must work consecutive weeks of maximum
7 weeks
	sommer[n,w1]+sommer[n,w2]<=1;

	Summerempl5{n in NS, s in Sw, w in Wsummer,d in WD[w]}:
	x[n,s,d] <= sommer[n,w];

# ALWAYS AT LEAST ONE NON-WEEKEND NURSE AND NON-SUMMEREMPLOYEE AT EACH SHIFT

	ExperienceLevel{s in Sw, d in SummerD}:
	sum{n in N:n not in NW and n not in NS}x[n,s,d]>=1;


#-----------------------------------HOLIDAYS------------------------------------


# MAXIMUM OF FOUR HOLIDAYS WITH WORK SHIFTS PER NURSE

	HDwork{n in N}:
	sum{d in HD}F3w[n,d]<=5;


# MAXIMUM OF SIX HOLIDAYS EACH WHEN INCLUDING THE NIGHT SHIFT FROM A HOLIDAY (2
HOURS OF HOLIDAY)

	HDwork2{n in N}:
	sum{d in HD}F3w[n,d]+sum{d in HD}x[n,"N",d]<=6;


# F3-DAY

	jobbHD{n in N, d in HD:n not in NS}:
	F3w[n,d]=x[n,"N",d-1]+x[n,"D",d]+x[n,"E",d];

	F3Dag1{n in N, d in HD: n not in NS}:
	F3w[n,d]+F3[n,d]=1;


#------------------------------FAIR DISTRIBUTION OF SHIFTS------------------------

# NIGHT SHIFTS DURING WEEKDAYS (MON-FRI)

	NumNightShifts{n in N}:
	sum{d in II}x[n,"N",d]=qn[n];

	NumNightShifts2{n in N100, nn in N100: n<>nn}:
	qn[n]=qn[nn];

	NumNightShifts3{n in N80, nn in N80: n<>nn}:
```

```
        qn[n]=qn[nn];

        NumNightShifts4{n in Neq, nn in Neq: n<>nn}:
        qn[n]=qn[nn];

        NumNightShifts5{n in N100, nn in N80}:
        qn[n]>=qn[nn];  #N100 nurses must work more night shifts than N80 nurses

        NumNightShifts6{n in N80, nn in Neq}:
        qn[n]>=qn[nn]+1; #N80 nurses must work more night shifts than Neq nurses

# EVENING SHIFTS DURING WEEKDAYS (MON-FRI)

        NumEveningShifts{n in N}:
        sum{d in II}x[n,"E",d]=qa[n];

        NumEveningShifts2{n in N100, nn in N100: n<>nn}:
        qa[n]=qa[nn];

        NumEveningShifts3{n in N80, nn in N80: n<>nn}:
        qa[n]=qa[nn];

        NumEveningShifts4{n in Neq, nn in Neq: n<>nn}:
        qa[n]=qa[nn];

        NumEveningShifts5{n in N100, nn in N80}:
        qa[n]>=qa[nn];  #N100 nurses must work more evening shifts than N80 nurses

        NumEveningShifts6{n in N80, nn in Neq}:
        qa[n]>=qa[nn]+1;  #N80 nurses must work more evening shifts than Neq nurses

# EQUAL DISTRIBUTION OF DAY AND EVENING SHIFTS DURING WEEKENDS

        D1D2{n in N: n not in NS and n <> "A24"}:
        D1[n]+D2[n]=sum{d in I}x[n,"D",d]-sum{d in I}x[n,"E",d];
```

Appendix 4: AMPL .run-**file** for Stage 1 and Stage 2 Calendar-Based Schedule

```
reset;
model CBScheduleStage1.mod;
data CBScheduleStage1.dat;
option solver gurobi;
option gurobi_options 'outlev=1 timelim=172800 logfile=MySavedLogFile1.log';
solve;

var T{N}; #Total weekends for nurse n
var TN{N}; #Total night weekends for nurse n
var Tvn{N}; #Total number of vacation weeks for nurse n

for {n in N}{
  let T[n] := sum{w in W}h[n,w];
  let TN[n] := sum{w in W}hn[n,w];
  let Tvn[n]:=sum{w in W}nv[n,w];
  }

display T, TN, Tvn;

# Creating an overview of weekends, night shift weekends and vacation weeks
assigned to each nurse

for {n in N}{
  print " " >Results_Stage1.txt;
  print "Nurse ", n >Results_Stage1.txt;
  printf "Helg: " >Results_Stage1.txt;
    for {w in W}{
      if h[n,w]==1 then
        printf "%1.0f,", w >Results_Stage1.txt; else
        continue;
     }
     print "" >Results_Stage1.txt;
     printf "Night Weekend: " >Results_Stage1.txt;
     for {w in W}{
     if hn[n,w]==1 then
     printf "%1.0f,",w >Results_Stage1.txt; else
     continue
     }
     print "" >Results_Stage1.txt;
     printf "Ferie: " >Results_Stage1.txt;
     for {w in W}{
     if nv[n,w]==1 then
     printf "%1.0f ", w >Results_Stage1.txt; else
     continue;
     }
     print "" >Results_Stage1.txt;
    }


# Creating parameters h, hn and nv and importing to .dat-file, so that they can be
used in Stage 2

#Param h
printf "param h: " >Text.dat;
for {w in W}{
  printf "%1.0f ",w > Text.dat;}
print ":=" >Text.dat;
```

```
for {n in N}{
printf "%s ", n >Text.dat;
 for {w in W}{
  printf "%1.0f ", h[n,w] >Text.dat;}
  print "">Text.dat;
  }
print ";" >Text.dat;


#Param hn
printf "param hn: " >Text.dat;
for {w in W}{
  printf "%1.0f ",w > Text.dat;}
print ":=" >Text.dat;

for {n in N}{
printf "%s ", n >Text.dat;
 for {w in W}{
  printf "%1.0f ", hn[n,w] >Text.dat;}
  print "">Text.dat;
  }
print ";" >Text.dat;


#Param nv
printf "param nv: " >Text.dat;
for {w in W}{
  printf "%1.0f ",w > Text.dat;}
print ":=" >Text.dat;

for {n in N}{
printf "%s ", n >Text.dat;
 for {w in W}{
  printf "%1.0f ", nv[n,w] >Text.dat;}
  print "">Text.dat;
  }
print ";" >Text.dat;


# Resetting and running stage 2 model
reset;
model CBScheduleStage2.mod;
data CBScheduleStage2.dat;
data Text.dat; # Parameter h, hn and nv from Stage 1
option solver gurobi;
option gurobi_options 'outlev=1 timelim=172800 logfile=MySavedLogFile2.log';
solve;


# Collecting data about which shifts as assigned to each nurse on each day, and
exporting to .txt-file.
# This is later imported to PyCharm and restructured using Python, before
importing it to Excel in same format as Haukeland uses for their schedules.
for {n in N, d in D} display {s in S: x[n,s,d]== 1} > nursesdfYEAR.txt;


# Creating .txt-file with contracted hours, scheduled hours and deviation for each
nurse
display E, y, z > nursehoursYEAR.txt;


# Defining variables for number of nurses assigned to each shift type on day d
var numnurseD{D};
var numnurseE{D};
var numnurseN{D};
```

```
for {d in D}{
 let numnurseD[d]:= sum{n in N}x[n,"D",d];
 let numnurseE[d]:= sum{n in N}x[n,"E",d];
 let numnurseN[d]:= sum{n in N}x[n,"N",d];
}

# Creating .txt-file with number of nurses on each shift
display numnurseD, numnurseE, numnurseN > numnurseYEAR.txt;

# Defining variables for number of each shift type for all nurses, and total
number of weekends
var Dayn{N};
var Evening{N};
var Night{N};
var WeekD{N};
var WeekE{N};
var WeekN{N};
var WeekendD{N};
var WeekendE{N};
var WeekendN{N};
var Totalh{N};
var Totalhn{N};


for {n in N}{
 let Dayn[n]:= sum{d in D}x[n,"D",d];
 let Evening[n]:= sum{d in D}x[n,"E",d];
 let Night[n]:= sum{d in D}x[n,"N",d];
 let WeekD[n]:= sum{d in II}x[n,"D",d];
 let WeekE[n]:= sum{d in II}x[n,"E",d];
 let WeekN[n]:= sum{d in II}x[n,"N",d];
 let WeekendD[n]:= sum{d in I}x[n,"D",d];
 let WeekendE[n]:= sum{d in I}x[n,"E",d];
 let WeekendN[n]:= sum{d in I}x[n,"N",d];
 let Totalh[n]:=sum{w in W}h[n,w];
 let Totalhn[n]:=sum{w in W}hn[n,w];
}


# Creating .txt-file with shift distributions
display Dayn, Evening, Night, WeekD, WeekE, WeekN, WeekendD, WeekendE, WeekendN >
shiftdistYEAR.txt;
```