University of Applied Sciences and Arts Northwestern Switzerland
School of Engineering

Universal Modelling and Optimization of Scheduling Problems

Bachelor Thesis

Student	*Elias Mjøen* *Joshua Schärer*
Expert	*Jonas Schwammberger*
Supervisor	*Simon Felix*
Client	-
Project number	IP6-22FS_14DS12
Agreement	-

University of Applied Sciences and Arts Northwestern Switzerland, School of Engineering

Windisch, August 19 2022

Abstract

Finding optimal solutions to scheduling problems is time-consuming, demands expertise in mathematical modeling, and often requires access to expensive solver software. This thesis discusses existing research made into the development of universal models and describes a new formulation for a universal model based on four base parameters: people, time slots, tasks, and locations. It is demonstrated how various constraints from a broad range of scheduling problems can be formulated over these constraints and how specific scheduling problems could be described with the proposed model.

As an alternative to solving the model with a standard ILP-Solver, an SAT-Solver is evaluated for scenarios of increasing complexity. The results show that the SAT-Solver can be used for small problems with negligible performance losses. For problems with increasing amounts of variables and constraints, the SAT-Solver starts to lose out on performance and can fail to produce results within acceptable time limits. It is nevertheless recommended to pursue the usage of SAT-Solvers as an alternative to ILP-Solvers for optimizing scheduling problems.

Table of Contents

List of Tables ..... **iv**

List of Figures ..... **iv**

List of Equations ..... **iv**

1 Introduction **1**

2 Background **3**

2.1 Literature about general models for scheduling problems *3*

2.2 Researched scheduling problems *4*

2.2.1 Nurse scheduling 4

2.2.2 Course scheduling 5

2.2.3 Examination scheduling 6

3 Formal definition of the universal model **7**

3.1 Notation *7*

3.1.1 Sets 7

3.1.2 Indexes 7

3.1.3 Optional Parameters 7

3.1.4 Decision Variables 8

3.2 Model *8*

3.3 Application of the model *14*

4 Implementing the model for end users **17**

4.1 Solvers *17*

4.2 Web Interface *17*

5 Testing and comparing the solvers **18**

5.1 Test overview *18*

5.2 Test metrics *19*

5.3 Test evaluation and discussion *20*

6 Conclusion **23**

List of references **24**

Declaration of academic honesty **25**

Appendix **26**

A All tests conducted for the nurse scheduling problem **26**

B All tests conducted for the course scheduling problem **27**

C All tests conducted for the examination scheduling problem **28**

List of Tables

Table 3.1: Representation of the nurse scheduling problem using the general model 14

Table 3.2: Representation of the course scheduling problem using the general model 15

Table 3.3: Representation of the examination scheduling problem using the general model 16

Table 5.1: Specifications of the testing environment 18

Table 5.2: Missing constraints in the implemented models for both the ILP-Solver and the SAT-Solver 18

Table 5.3: Test metrics against which the tests are measured 19

List of Figures

Figure 1.1: Illustration of the calculated result of the examination scheduling problem using the ILP-Solver on the web application. 1

Figure 2.1: Sketch of a timetable for solutions produced with our general model 3

Figure 5.1 Comparision of large integer values for both the ILP-Solver and the SAT-Solver 20

Figure 5.2: Comparison of the consistent performance for both the ILP-Solver and SAT-Solver 21

Figure 5.3 Comparision of the peak RAM usage (GB) for both solvers 22

List of Equations

Equation 3.1 8

Equation 3.2 9

Equation 3.3 9

Equation 3.4 9

Equation 3.5 9

Equation 3.6 9

Equation 3.7 9

Equation 3.8 10

Equation 3.9 10

Equation 3.10 10

Equation 3.11 11

Equation 3.12 11

Equation 3.13 11

Equation 3.14 11

Equation 3.15 12
Equation 3.16 12
Equation 3.17 12
Equation 3.18 12
Equation 3.19 13
Equation 3.20 13
Equation 3.21 13
Equation 3.22 13
Equation 3.23 13

1 Introduction

Scheduling problems are a subset of optimization problems that involve the scheduling of a number of tasks over a given time period under given objectives and constraints. Scheduling problems appear in various environments and domains. One of the more common and well-known scheduling problems is the scheduling of personnel/employees in any company with multiple workers. Simple models can often be solved by hand on a whiteboard or piece of paper, but with increasing complexity, this approach becomes less realistic. In practice, the more complex problems are solved with the help of computer programs called solvers that can operate on millions of variables and constraints.

Many applications and models for specific scheduling problems have been created and published but adapting these solutions to different scheduling problems is often more complicated and time-consuming than developing new models for them. Therefore, we propose a model that is flexible enough to accommodate a range of scheduling problems and provide a class library written in C# .Net that implements our model and demonstrates operation on multiple examples of scheduling problems. We also evaluate how solvers, specifically an Integer Linear Programming (ILP) solver and a CDCL-based SAT-Solver, perform in solving various scenarios under our model.

This thesis also includes the creation of a blazor web application that sits on top of a library implementing the universal model. The user may utilize this interface to enter their own desired scheduling problems, using all the features provided by the general model. The designed scheduling problem can then be computed using either the ILP-Solver or the SAT-Solver. Following Figure 1.1 shows such a result of the examination scheduling problem displayed on the web application.

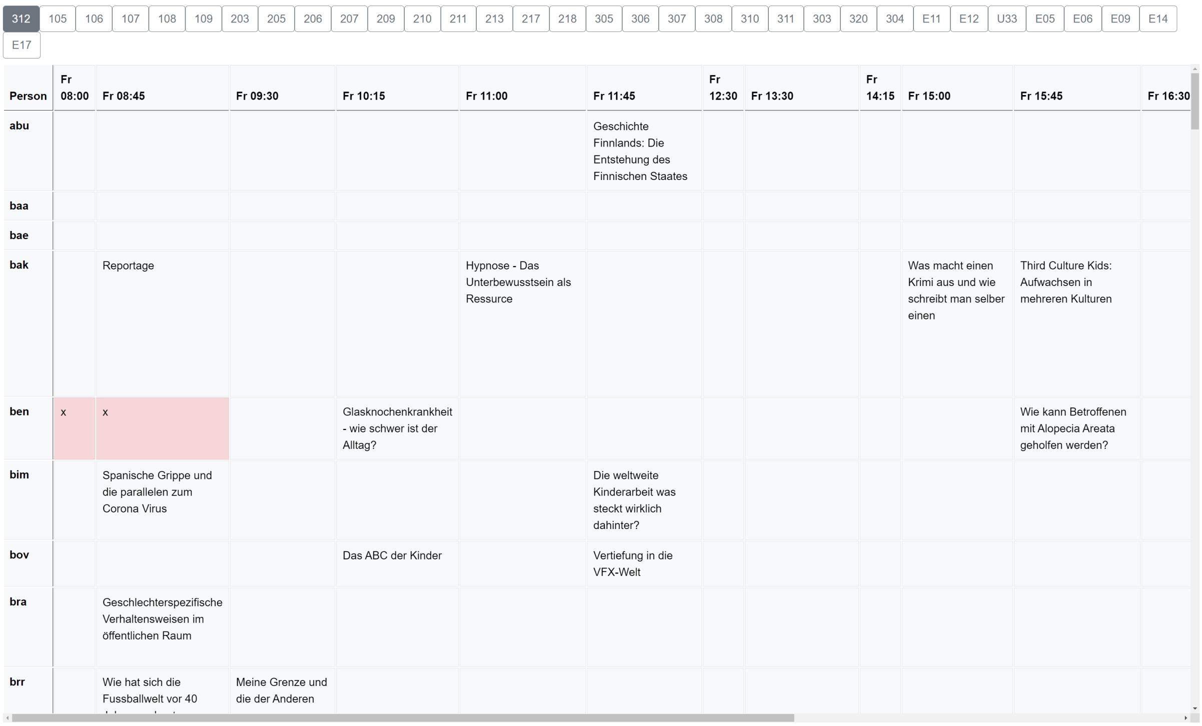

Figure 1.1: Illustration of the calculated result of the examination scheduling problem using the ILP-Solver on the web application.

Well, visible in this figure is the representation of the computed result in the form of a timetable. The columns display the time slots and the rows represent the people. The assigned tasks, in this case the presentation, appear as cells. Additionally, the locations are changeable all the way at the top.

Chapter 2 presents scheduling problems by looking into existing research and defining which kinds of scheduling problems this paper focuses on. In parallel, the chapter discusses research into these problems and provides a closer look at how different kinds of constraints can be modeled. In chapter 3 we formally propose a universal model mathematically and discuss how different types of scheduling problems can be represented with the general model. Chapter 4 proceeds to explain the implementation of our model for the end user. Chapter 5 compares both the ILP-Solver and the SAT-Solver using test results and test metrics. Finally, chapter 6, provides a conclusion outlining all the key findings.

2 Background

While there are countless variations of scheduling problems, most of them share some characteristics and are often classified based thereon[1]. It is important to state that even though it is possible to identify such characteristics in theory, in practical scenarios such classification is of ambiguous nature since many real-world problems often deviate from their classification with additional complexities like unique constraints and specialized variables that originate from their domain and field of application.

2.1 Literature about general models for scheduling problems

There have been propositions for general models for staff scheduling problems before, such as the model proposed in [2] or the one proposed for timetabling problems in [3]. While the model for staff scheduling problems was developed by extending a model based on one scenario with parameters and constraints of others, the model for timetabling problems was developed by first researching and compiling a number of commonly found constraints, variables and parameters into a model which was then tested in various scenarios.

[4] introduced classification for personnel scheduling problems based on three base parameters: personnel (hereafter referred to as person in singular or people in plural), time (hereafter referred to as time slots), and duties (hereafter referred to as tasks). Additional parameters for qualifications (hereafter referred to as skills) of people for tasks and another for locations are also mentioned but not elaborated further. Additionally, they describe the solution space for researched personnel scheduling problems as having multiple dimensions. We observed that in most of the considered problem scenarios, these dimensions are equivalent to the axes found in timetables, the most common form of visualization used for schedules. The approach described in [4] could be extended by adding a fourth dimension for cases where the location parameter is needed. The resulting solution space can then be visualized as timetables as in Figure 2.1.

Figure 2.1: Sketch of a timetable for solutions produced with our general model

2.2 Researched scheduling problems

The following problems served as basis to our generalization. The problems parameters and most of their constraints as well as their objective(s) are textually described, and later formally defined as scenarios under our model in chapter 3. Constraints are divided into soft and hard constraints. Hard constraints define solution feasibility, meaning solutions that violate a hard constraint are considered infeasible. Soft constraints define solution quality, meaning solutions that violate a soft constraint are of worse quality than ones that do not.

Each of the problems was subject to previous research and there exist solutions and documentation for each of them, which makes the process of evaluating the model in the context of these problems easier.

2.2.1 Nurse scheduling

The first problem to be addressed is that of nurse rostering. This is done using the following scenario as described in the thesis [5].

Type of scheduling problem:	Nurse scheduling, nurse rostering
Description:	In a Swiss hospital, assistant doctors need to be assigned to shifts over the period of a month. The produced timetable should help in bringing down labor costs and providing fairer work shift distribution for the doctors.
People:	Assistant doctors
Time slots:	Days (usually one month)
Tasks:	Night shift, early shift, late shift
Locations:	One hospital
Hard constraints:	Night shifts need to be scheduled over seven days and are compensated with one off-day before and two off-days after the night shift block.
	Doctors can be assigned at most one shift per day
	Shifts on weekends need to be compensated during the week prior with one off-day per workday on weekends.
	Because of regulatory restrictions, early shifts cannot follow directly after late shifts
Soft constraints:	The shifts should be distributed fairly across doctors
Objective(s):	Minimize labor costs, maximize fairness

Nurse rostering is a well-known staff scheduling problem found in hospitals and is known to be of NP-Hard complexity. *Cheang et al.* [6] give a good overview of common variables and constraints found in nurse scheduling problems and also propose a few approaches to solving the problem. The scenario above serves as an example of nurse scheduling in this thesis.

2.2.2 Course scheduling

The second type of scheduling problem is course scheduling.

Type of scheduling problem:	General staff scheduling problem, course scheduling problem
Description:	A company that offers various courses for further education needs to schedule speakers for their courses every year. The dates and locations for the courses are fixed beforehand and the courses need certain skills to teach. Speakers have travel costs to each of the locations which are paid for by the company. Additional external speakers can be accessed at a much higher cost.
People:	Speakers
Time slots:	Days (dates over one year)
Tasks:	Courses with one fixed location and multiple fixed dates
Locations:	Rooms in facilities across Switzerland
Hard constraints:	Speakers can only talk in one location per day
	Speakers have topics which they can teach (skills) that correspond to course topics.
	Courses have multiple topics
	The optimal number of distinct speakers per course is 2
Soft constraints:	Speakers defined workloads should be met
Objective(s):	Minimize labor and travel costs

The above-described scenario is the largest and most complex out of the three scenarios and is also the hardest to properly categorize. The problem shows characteristics of general staff scheduling problems with people needing to be assigned to certain tasks, but with the courses having fixed dates and fixed locations, the challenge is optimally assigning available speakers to the locations. Speakers having travel costs associated with every location indicates characteristics of crew scheduling or even the traveling salesman problem.

2.2.3 Examination scheduling

The third and last evaluated problem is that of examination scheduling. Thesis [7] provides a realistic scenario for this type of scheduling problem.

Type of scheduling problem:	Examination scheduling problem
Description:	A school that has presentations during the last two school days every year needs to assign these presentations to rooms. Some presentations have requirements for rooms, such as presentations of dances requiring a room that provides a floor that can be danced on.
People:	Teachers, experts and students
Time slots:	45-minute time slots over multiple days.
Tasks:	Presentations with disciplines (skills)
Locations:	Rooms with facilities for disciplines (skills)
Hard Constraints:	The presentations have disciplines that belong to a category (normal, music, dance, art) and presentations can only be held in a room with the appropriate facilities. Students need to watch at least three presentations of other students. A presentation needs the presence of the presenting student, a teacher, and an expert and optionally a student audience.
Soft constraints:	The number of room changes for a teacher should be minimized. Teachers should have as few empty time slots in between as possible. Students can name preferences for presentations they want to watch.
Objective(s):	Minimize planning period, minimize room cost

This scenario is a textbook example of the examination scheduling problem that is commonly found in universities and other educational institutions. One simplification is that the teachers are already being assigned as coaches and experts to the presentations beforehand.

3 Formal definition of the universal model

A new mathematical formulation of a universal model is created, based on the observations made in the previous chapter, which we propose as a universal model for scheduling problems. In the following paragraphs, we formally define the model in mathematical terms and then express the previously described scheduling problems with these formulations. We thereby prove that the model indeed has universal traits and can easily be applied to other scheduling problems.

3.1 Notation

The main variables or base parameters are described in sets. As opposed to other researched papers [3], which propose a universal model using only three different variable sets, this thesis proposes a general model including four distinct variables.

While some scenarios may require time to be continuous, we decided to use timeslots. This simplification is still flexible enough to cover all of the considered scheduling problems while allowing the model to use binary decision variables for all base parameters. Some scheduling problems may only require two or three of the base parameters. For example, a nurse scheduling problem may consider shifts/tasks to be time constrained already, in which case the unused time parameter should be set to one.

3.1.1 Sets

- *People* $\boldsymbol{P}$: Set of distinct people. P can be short for |P|
- *Time slots* $\boldsymbol{T}$: Set of distinct time slots. T can be short for |T|
- *Tasks* $\boldsymbol{I}$: Set of distinct tasks. I can be short for |I|
- *Locations* $\boldsymbol{L}$: Set of distinct Locations. L can be short for |L|

3.1.2 Indexes

- *Person* $\boldsymbol{p} \in \{0, \dots ,\boldsymbol{P} - 1\}$
- *Time slot* $\boldsymbol{t} \in \{0, \dots ,\boldsymbol{T} - 1\}$
- *Task* $\boldsymbol{i} \in \{0, \dots ,\boldsymbol{I} - 1\}$
- *Location* $\boldsymbol{l} \in \{0, \dots ,\boldsymbol{L} - 1\}$

3.1.3 Optional Parameters

The following parameters are used in addition to the base parameters in the scheduling problems that were described earlier. They are meant to demonstrate how a wide range of constraints could be formulated over the defined sets and are used as examples for common constraints.

Soft constraint parameters:

- Wage $\boldsymbol{W_p}$: Cost for person p to work per time slot
- Travel costs $\boldsymbol{K_{l,p}}$: $\boldsymbol{L} \times \boldsymbol{P}$: Cost for person p to travel to location l
- Workload $\boldsymbol{G_p}$**:** Percentage of total time slots (T) in which person p should work
- Preferred number of people $\boldsymbol{U_i}$: Preferred number of different people per task i
- Weighting $\boldsymbol{M} =$ {wage, travel cost, workload, location changes, number of empty time slots, number of used timeslots, number of used locations, fairness}: number.

Hard constraint parameters:

- Absences $\boldsymbol{A_{p,t}}$: $\boldsymbol{P} \times \boldsymbol{T}$: 1 if person p is absent at time slot t, 0 otherwise
- Capacity $\boldsymbol{C_p}$**:** Number of tasks person p can be assigned to in one time slot
- Repetitions $\boldsymbol{R_i}$: Number of occurrences of task *i* over the planning period
- Consecutive time slots $\boldsymbol{H_i}$: Number of consecutive time slots of task *i*
- Required number of people $\boldsymbol{V_{i,t}}$: $\boldsymbol{I} \times \boldsymbol{T}$: Minimum number of people required for task *i* in time slot t
- Required people $\boldsymbol{QP_{i,p}}$: $\boldsymbol{I} \times \boldsymbol{P}$: 1 if task *i* needs to be assigned to person p, 0 otherwise
- Required time slots $\boldsymbol{QT_{i,t}}$: $\boldsymbol{I} \times \boldsymbol{T}$: 1 if task *i* needs to be assigned in time slot *i*, 0 otherwise
- Required locations $\boldsymbol{QL_{i,l}}$: $\boldsymbol{I} \times \boldsymbol{L}$: 1 if task *i* needs to be assigned to location *l*, 0 otherwise
- Skills ***S:*** Set of skills that can be assigned to people, tasks, and locations
- Personal skills $\boldsymbol{SP_{s,p}}$: $\boldsymbol{S} \times \boldsymbol{P}$: 1 if person p has skill *s, 0 otherwise*
- Locational skills $\boldsymbol{SL_{s,l}}$: $\boldsymbol{S} \times \boldsymbol{L}$: 1 if location l provides facilities for skill *s*, 0 otherwise
- Task skills $\boldsymbol{SI_{s,i,t}}$: $\boldsymbol{S} \times \boldsymbol{I} \times \boldsymbol{T}$: 1 if skill *s* is needed for task *i* in timeslot *t*, 0 otherwise
- Compensation before: $\boldsymbol{CO_i^{before}}$: Tuple(0: Number of time slots to compensate before the one in which task *i* is assigned, 1: Number representing the range of timeslots before the assignment during which the task assignment has to be compensated.)
- Compensation after: $\boldsymbol{CO_i^{after}}$: Tuple(0: Number of time slots to compensate after the one in which task *i* is assigned, 1: Number representing the range of timeslots after the assignment during which the task assignment has to be compensated.)
- Blocked tasks: $\boldsymbol{BL_i^{after}}$**:** Set of tasks that cannot be assigned following an assignment of task *i*.

3.1.4 Decision Variables

Binary decision variables are formulated as follows and are henceforth used.

- $\boldsymbol{X}_{p,t,i,l} = \begin{cases} 1 \equiv TRUE, & if\ p\ is\ assigned\ to\ i\ at\ t\ in\ l \\ 0 \equiv FALSE, & otherwise \end{cases}$

Some implementations may use other forms of decision variables such as sets of integers like:

- $\boldsymbol{X}_{p,i,l} =$ List of time slot indexes at which p is assigned to *i* in *l*

3.2 Model

Soft constraints:

Equation 3.1
$$c_0 = \sum_p^P \sum_t^T \sum_i^I \sum_l^L X_{p,t,i,l} \cdot W_p$$
Wage

This constraint describes the labor costs for every time slot that every person is working. Since all three scheduling problems involve labor costs for people it must be part of the general model. They are deliberately designated as wages, as this is the most common way of modeling labor costs.

In addition, labor costs for time slots, tasks, or even locations are conceivable as well. However, since none of the three types of scheduling problems require any of these, we choose to forego any further examples of such similar constraints.

Equation 3.2 $$c_1 = \sum_p^P \sum_t^T \sum_i^I \sum_l^L X_{p,t,i,l} \cdot K_{l,p}$$ Travel costs

The equation c_1 above states the cost of travel for every person to their assigned locations. This constraint originates specifically from the course scheduling problem, where the speakers' travel costs are paid for by the company.

Equation 3.3 $$c_{2.1} = \left| \left(\left(\sum_t^T \bigvee_i^I \bigvee_l^L X_{p,t,i,l} \right) \cdot \frac{100}{T} \right) - G_p \right|, \qquad \forall\, p$$ Workload

Equation 3.4 $$c_2 = max\{c_{2.1}\} - min\{c_{2.1}\}$$ Fairness

$c_{2.1}$ imposes a penalty on over- or under-stepping the people's contractual workload by adding up the differences between a person's actual and intended workloads. C_2 then ensures that the deviations from every person's workload are distributed equally and thereby that the work is divided fairly among all people.

This constraint is inspired and derived by this thesis [5]. However, different from what is described in the paper, the general model proposed in this theorem includes a more broadly formulated fairness constraint. Different types of scheduling problems have different perceptions of the fairness constraint. Specific real-world problems further complicate the matter. To make the model universal and easily usable for a wide range of scheduling problems, we decided to compromise on precision and use only the workload constraint for fairness.

Equation 3.5 $$c_3 = \sum_p^P \sum_{t=1}^T \sum_l^L \left(\bigvee_i^I X_{p,t,i,l} \oplus \bigvee_i^I X_{p,t-1,i,l} \right) \wedge \left(\bigvee_i^I X_{p,t,i,l} \right)$$ Number of location changes

This equation c_3 accumulates the number of location changes every person makes. It is derived from the same constraint formulated in the thesis [7].

Although it is used in only one of the three problems, it is still worthwhile incorporating it into the general model since it can be easily adapted to countless other scheduling problems.

Equation 3.6
$$t_{last} = \begin{cases} 1, \left(\sum_{t+1}^T \bigvee_i^I \bigvee_l^L X_{p,t,i,l} = 0 \right) \wedge \left(\bigvee_i^I \bigvee_l^L X_{p,t,i,l} = 1 \right) \\ 0, otherwise \end{cases}, \ \forall\, p$$
$$t_{first} = \begin{cases} 1, \left(\sum_0^t \bigvee_i^I \bigvee_l^L X_{p,t,i,l} = 0 \right) \wedge \left(\bigvee_i^I \bigvee_l^L X_{p,t,i,l} = 1 \right) \\ 0, otherwise \end{cases}, \ \forall\, p$$
First and last assigned time slot

Equation 3.7 $$c_4 = T - \left(\sum_t^T \bigvee_i^I \bigvee_l^L X_{p,t,i,l} \right) - t_{first} + (T - t_{last}), \qquad \forall\, p$$ Number of empty time slots

This soft constraint adds up the empty time slots between the first time slot t_{first} and last time slot t_{last} for every person. It originates from the examination scheduling problem as well and is also a derivation of the one formulated in the thesis [7].

Similar to equation 3.6, this unique constraint is also included in the model to increase the overall universality, since we see countless other applications of this constraint in several other types of scheduling problems.

Equation 3.8

$$c_5 = \sum_t^T \bigvee_p^P \bigvee_i^I \bigvee_l^L X_{p,t,i,l}$$

Number of used time slots

Equation 3.9

$$c_6 = \sum_l^L \bigvee_p^P \bigvee_i^I \bigvee_t^T X_{p,t,i,l}$$

Number of used locations

The above equations c_5 and c_6 sum on one hand all the used time slots and on the other all the used locations in which people have tasks. Both soft constraints originate from the examination scheduling problem.

Equation 3.10

$$c_7 = \left| \left(\sum_p^P \bigvee_t^T \bigvee_l^L X_{p,t,i,l} \right) - U_i \right|, \qquad \forall\, i$$

Preferred number of people

c_8 intends to assign a preferred amount of different people to each task. This soft constraint originates from the course scheduling problem.

Despite being essential only to the course scheduling problem, it is nevertheless highly beneficial to incorporate it into the general model, as it is quite easy to apply it to other types of scheduling problems.

Formal definition of the universal model

Minimize:

Equation 3.11

$$\sum_{n=0}^{M} M_n \cdot c_n$$

Objective

Describes the objective function. The objective function tries to minimize a weighted sum of the results of each soft constraint (Equations 3.1 – 3.10). The intention of the weights is to have them be based on cost. Additionally, specific soft constraints can be excluded from the model by setting the according weight in M to 0. Allowing the general model to minimize only those soft constraints that are necessary for the desired scheduling problem.

Subject to:

Equation 3.12

$$\sum_{i}^{I} \sum_{l}^{L} X_{p,a,i,l} = 0, \qquad \forall\, p, a \in A_p$$

Absences

Allows for the definition of availability of people. The model defines availability as a set of time slots where the person is not available, also referred to as a blacklist. Another option would be to define it as a whitelist, meaning a set of time slots where the person is available. We did not go for that option because for the majority of real-world scenarios it is assumed that contracted people are available most of the time and not vice-versa. Another reason is that the parameter for absences can be left empty as default, if the constraint should not be considered, instead of having to supply the total list of timeslots as default.

This hard constraint is common in a range of different types of scheduling problems, and thus needs to be included in the general model.

Equation 3.13

$$\sum_{i}^{I} \sum_{l}^{L} X_{p,t,i,l} \leq C_p, \qquad \forall\, p, t$$

Capacity

The above equation 3.13 describes the constraint of having people be able to work on multiple tasks in one time slot. The analysis of different scheduling problems indicated that this is generally not the case and that a person is usually expected to work on only one task per time slot. However, this formulation is not only broader in scope, but also allows the model to describe the externally recruited speakers as one person, rather than including multiple people in the course scheduling problem.

Equation 3.14

$$\sum_{t}^{T} \bigvee_{p}^{P} \bigvee_{l}^{L} X_{p,t,i,l} \geq R_i\,, \qquad \forall\, i$$

Repetitions

Describes the number of times a task needs to be repeated over the entire planning period. This constraint allows a lot of flexibility when describing tasks, as it eliminates the need to define similar or repeating tasks in separate instances. If the corresponding parameter R_i is set to one, the constraint will not be considered.

Equation 3.15

$$\bigvee_{l}^{L} X_{p,t,i,l} \Rightarrow \sum_{t}^{H_i} \bigvee_{l}^{L} X_{p,t,i,l} = H_i\,, \qquad \forall\, p,t,i$$

Consecutive time slots

The hard constraint above formulates the amount of time slots a task needs to be repeated consecutively. It originates from the nurse scheduling problem and pursues the objective of assigning tasks i, described as shifts in the nurse scheduling, over a number of consecutive time slots t, here days.

Equation 3.16

$$\bigvee_{l}^{L} X_{p,t,i,l} \Rightarrow \sum_{t-CO_{i,1}^{before}}^{t} \bigvee_{i}^{I} \bigvee_{l}^{L} X_{p,t,i,l} = \left(CO_{i,1}^{before} - O_{i,1}^{before}\right)$$

$$\forall\, p,t,i$$

$$\bigvee_{l}^{L} X_{p,t,i,l} \Rightarrow \sum_{t+1}^{CO_{i}^{after}+1} \bigvee_{i}^{I} \bigvee_{l}^{L} X_{p,t,i,l} = \left(CO_{i,1}^{after} - O_{i,1}^{after}\right),$$

$$\forall\, p,t,i$$

Compensation before and after certain tasks

Describes the amount of time slots t for every person working on the task i that should be compensated beforehand CO_i^{before} as well as the amount that should be compensated afterwards CO_i^{after}.

This is a fairly common hard constraint in the nurse scheduling problem. Assistant doctors may require a few time slots off directly before and after a certain task. The universal model incorporates an even extended formula which also allows to specify a range of time slots in which the assignments need to be compensated. Paper [5] describes this kind of weekend time slots, which need to be compensated for the working person p a week prior.

Equation 3.17

$$\bigvee_{l}^{L} X_{p,t,i,l} \Rightarrow \left(\bigvee_{l}^{L} X_{p,t+1,i\,\in BL_i^{after},l}\right), \qquad \forall\, p,t,i$$

Blocked tasks after certain tasks

It is not allowed for the person p to be assigned to blocked tasks BL_i^{after} directly on the following time slot t for task i.s

Equation 3.18

$$\sum_{p}^{P} X_{p,t,i,l} = V_i, \qquad \forall\, t,i,l$$

Required number of people

Describes the number of people that are required in time slot t to complete task i. The observation of various types of scheduling problems reveals that this hard condition is fairly common in a large number of them. The parameter V_i is defaulted to one for problems that do not model this constraint.

Formal definition of the universal model

Equation 3.19	$$\sum_{t}^{T}\sum_{l}^{L} X_{p,t,i,l} \geq QP_{i,p}, \qquad \forall\, i, p$$	Required people
Equation 3.20	$$\sum_{p}^{P}\sum_{l}^{L} X_{p,t,i,l} \geq QT_{i,t}, \qquad \forall\, i, t$$	Required time slots
Equation 3.21	$$\sum_{p}^{P}\sum_{t}^{T} X_{p,t,i,l} \geq QL_{i,l}, \qquad \forall\, i, l$$	Required locations

These three hard constraints allow the assignment of specific people $QP_{i,p}$, time slots $QT_{i,t}$ and locations $QL_{i,l}$ to every task i. This condition is applicable in the course scheduling problem as well as in the examination scheduling problem.

Equation 3.22	$$X_{p,t,i,l} \leq SP_{s,p} \wedge SI_{s,i,t}, \qquad \forall\, p, t, i, l, s$$	Personal skills
Equation 3.23	$$X_{p,t,i,l} \leq SL_{s,l} \wedge SI_{s,i,t}, \qquad \forall\, p, t, i, l, s$$	Locational skills

Describes hard constraints that limit which people and locations can be assigned to tasks. Only people with the same set of skills $SP_{s,p}$ as well as locations with the same set of skills $SL_{s,l}$ as the ones required by the task $SI_{s,i,t}$ are allowed to be assigned to the task. Leaving the set of skills S empty excludes both constraints from the general model. Moreover, can only one of the distinct parameters $SP_{s,p}$, and $SL_{s,l}$ be left empty, which would lead to the corresponding constraints being excluded.

These kinds of constraints appear in all three types of scheduling problems studied in this thesis. In contrast to what is described in the thesis [7], for example, our model constitutes a more universal representation. Similar to what is described in the theorem, our model also incorporates a required type of location for a task in that regard. However, this universal model takes it a step further and extends the same assignment to the person as well, linking all three variables together through one set of skills. This approach combines both limitations in one general set of skills.

3.3 Application of the model

This chapter demonstrates how the problems described in section 2.2 can be represented with the proposed general model. The following three tables outline how each problem can be represented using the defined sets, parameters, and constraints of the model.

Table 3.1: Representation of the nurse scheduling problem using the general model

Sets		Notation
Total number of assistant doctors		P
Total number of days		T
Total number of shifts		S
Total number of rooms		L
Constraints	Equations	Parameter
All assistant doctors achieve their workload to the best of their ability	Equation 3.3	G_p
The shifts are distributed fairly across all assistant doctors	Equation 3.4	M
Absences of the assistant doctors	Equation 3.12	$A_{p,t}$
Assistant doctors work at most one shift per day	Equation 3.13	C_p
Night shifts are scheduled over seven consecutive days	Equation 3.15	H_i
Assistant doctors have one day off before and after the night shift	Equation 3.16	$CO_{p,i}^{before}$ $CO_{p,i}^{after}$
Weekend shifts are compensated during the prior week with one day off for each workday on the weekend	Equation 3.16	$CO_{p,i}^{before}$
Assistant doctors can not work early shifts directly after late shifts	Equation 3.17	BL_i^{after}

Formal definition of the universal model

Table 3.2: Representation of the course scheduling problem using the general model

Sets		Notation
Total number of speakers		P
Total number of half days		T
Total number of courses		S
Total number of locations		L
Total number of topics		S
Constraints	Equations	Parameter
All speakers have a wage	Equation 3.1	W_p
Speakers face varying travel costs from them to the locations	Equation 3.2	$K_{l,p}$
Speakers defined workloads should be met	Equation 3.3	G_p
The optimal number of distinct speakers per course is 2	Equation 3.10	U_i
Absences of speakers	Equation 3.12	$A_{p,t}$
Speakers can only talk in one location per day	Equation 3.13	C_p
Courses are repeated for a certain number of days	Equation 3.14	R_i
Courses may require specific days and locations	Equation 3.20 Equation 3.21	$QT_{i,t}$ $QT_{i,l}$
Courses are based on topics that can only be taught by a speaker who is familiar with the same topics	Equation 3.22	$SP_{s,p}$ $SI_{s,i,t}$

Table 3.3: Representation of the examination scheduling problem using the general model

Sets		Notation
Total number of teachers and experts		P
Total number of 45 minutes lectures		T
Total number of presentations		S
Total number of rooms		L
Total number of topics		S
Constraints	Equations	Parameter
The number of room changes for a teacher should be minimized	Equation 3.5	M
Teachers should spend as little time as possible between presentations	Equation 3.7	M
Use overall as little room and time slots as possible.	Equation 3.8 Equation 3.9	M
Absences of teachers and experts	Equation 3.12	$A_{p,t}$
A presentation requires the presence of the presenting student, the teacher, and the expert	Equation 3.18 Equation 3.19	$V_{i,t}$ $QP_{i,p}$
Presentations are on different topics and can only take place in a room with the corresponding facilities.	Equation 3.23	$SL_{s,l}$ $SI_{s,i,t}$

An important finding is that all of the problems utilize all four sets described in chapter 3.1.1. However, each problem uses these sets differently. For example, the persons range from assistant doctors to teachers. Or also the tasks represent sometimes shifts, which reach over several timeslots, up to presentations which take place only once at a time. Further, can the timeslots represent whole days like in the nurse scheduling problem or only single 45 minutes lessons as described in the examination scheduling problem. The same applies to the parameters and constraints. In many cases, these are used repeatedly over several different problems, such as the absences, but specific constraints, such as the preferred number of different people per task, as required in the course scheduling problem, can also be mapped with the general model. Finally, it can be confirmed that each of the three problems can be easily represented with the model.

4 Implementing the model for end users

As a proof of concept, the universal model proposed in this thesis was implemented in a C# class library in .NET 6.0. The library provides documented classes, involving parameters and variables, as well as an interface for solvers. On top of the class library, the scope of this thesis also involves the creation of a blazor web application that serves as an interface for end users. The interface allows the input of new scheduling problems, access to real-world examples, and computes solutions for the desired scheduling problems using the provided solver implementations.

4.1 Solvers

This thesis focuses on two distinct solvers: The gurobi optimizer is a well-known and highly optimized mathematical solver. It provides algorithms for solving ILP-Programs, among others. To implement the proposed model for the gurobi solver we used the .NET interface that is available for free with an academic license. The .NET Library SATInterace [8] is used to implement our model with the CaDiCal SAT-Solver.

The universal model is implemented identically for both solvers. The decision variables are represented by a four-dimensional array incorporating people, time slots, tasks, and locations. The benefits of this approach are the arrays ease of access and its similarities to the decision variables proposed in the general model in chapter 3.1. The drawbacks are its resource usage, especially the amount of memory needed and the need to iterate over all four dimensions every time multiple variables need to be accessed.

An alternative is splitting the array into multiple smaller arrays and referencing them to each other. However, that option was not implemented, because of the complexity arising when formulating constraints with that approach.

Further details on the evaluation and the comparison of both solvers follow in chapter 5.

4.2 Web Interface

The web interface is designed rather simply. It represents the universal model exactly, allowing inputs of sets and parameters.

It is a very interesting question as to how to properly present complex information like this to an uninitiated person, which would provide enough material for another paper. However, answering this question is not within the scope of this project, so the design is kept plain and simple.

5 Testing and comparing the solvers

In this chapter, the scenarios described in chapter 3.3 are applied to both the ILP-Solver and the SAT-Solver. The results are measured, documented, and evaluated based on defined metrics.

5.1 Test overview

The model is validated through various tests. The second goal of this thesis is to analyze whether the SAT-Solver is a suitable alternative to the ILP-Solver for solving scheduling problems. For that reason, the two solvers are inspected on a variety of aspects in the following chapters.

All measurements are performed in the same test environment under the same conditions. Thus, allowing all results to be in relation to each other. **Error! Reference source not found.** shows the specifications used in our test environment. Another important note is that all tests are performed in the release build of our .net application in order to increase the overall performance.

Table 5.1: Specifications of the testing environment

Processor	Intel(R) Core(TM) i9-7900X CPU @ 3.30GHz, 10 Core(s), 20 Logical Processor(s)
Physical memory (RAM)	32.0 GB

An additional remark is that due to time restrictions, not all constraints proposed in our general model could be implemented for both the ILP-Solver and the SAT-Solver. However, both models for the solvers support identical constraints, which allows the comparison, nevertheless. Table 5.2 lists all the missing constraints for the related scheduling problem.

Table 5.2: Missing constraints in the implemented models for both the ILP-Solver and the SAT-Solver

Solver	Missing constraints	
	Nurse scheduling	Examination scheduling
ILP	Equation 3.6, Equation 3.8	Equation 3.9-10, Equation 3.16-19
SAT	Equation 3.6, Equation 3.8	Equation 3.9-10, Equation 3.16-19

In summary, the majority of these constraints are primarily performed in the nurse scheduling problem. Consequently, the comparison of the two solvers for this problem is not fully reflecting reality, but it is still in relation to each other. The same applies to the examination scheduling problem.

5.2 Test metrics

This section describes the metrics used in interpreting the test results. These metrics assist in comparing both solvers objectively with one another and help ensure that all measurements are both realistic and reproducible.

In addition, Table 5.3 lists all the metrics against which the tests are subsequently measured.

Table 5.3: Test metrics against which the tests are measured

Reference	Test metric	Description
TM1	Optimality	Each scenario is divided into three groups with different amounts of data: • Small: Data set with 10’000 decision variables [10 people, 10 time slots, 10 tasks, 10 locations, 5 skills, and ~1’000 constraints] • Medium: Data set with 1’000’000 decision variables [20 people, 50 time slots, 100 tasks, 10 locations, 10 skills, and ~10’000 constraints] • Big: Data set with 10’000’000 decision variables [50 people, 50 time slots, 200 tasks, 20 locations, 20 skills, and ~100’000 constraints] Each solver is performed under appropriate time restrictions for the complexity of the scenario: • Small: 10s, 30s, 1m • Medium: 5m, 15m • Big: 30m, 1h The solver providing the cheapest solution within the time limits is counted as the better one for that scenario. Thus, performance is measured by this metric.
TM2	Scalability	Each of the solvers is evaluated inside the complexity classes to find out how well it scales with time. This is measured by the improvement of the objective if given additional time. Additionally, the solvers are evaluated on how well they scale across complexity classes by examining the tendencies shown when increasing complexity across all scenarios.
TM3	Consistency	In order to evaluate the stability of the solution, each solver is executed three times. Performing similarly over several runs using different seeds verifies the consistency of each solver.

5.3 Test evaluation and discussion

This chapter highlights interesting observations made during the test assessment. Indicative findings are picked from the tables in the appendix and are then analyzed and evaluated in a brief discussion. Conclusions are then drawn in consideration of the test metrics specified in Table 5.3. Detailed test results can be viewed in tables A, B, and C attached in the appendix.

One first interesting observation is found while evaluating the optimality of each solver. As described in Table 5.3, TM1 Optimality serves as a performance reference. The quality of the results is assessed in terms of cost. The solver with the cheaper solution within the given time limit is to be preferred. Analyzing all gathered test results show that the ILP-Solver almost always computes the best possible solution for each complexity class within the given time limit. Only for the nurse scheduling problem does the ILP-Solver not come up with the highest quality solution from the medium complexity onwards. Whereas the SAT-Solver, with the exception of the examination scheduling problem, runs into time limitations and is therefore not able to come up with the most optimal solution. In some cases found in the course scheduling problem, the SAT solver is even unable to find a solution at all, in the given time restriction. The scenario for course scheduling problems produces large integer objectives and the SAT-Solver seems to struggle with that, as it tries to optimize solutions in a very large interval and struggles to exempt suboptimal solutions as fast as the ILP-Solver. The following Figure 5.1 illustrates exactly this behavior.

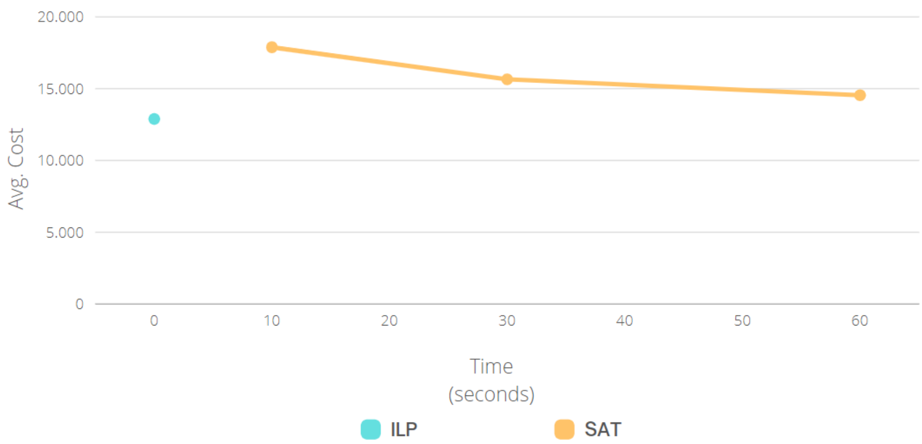

Figure 5.1 Comparision of large integer values for both the ILP-Solver and the SAT-Solver

This makes sense as SAT-Solvers usually operate solely on boolean variables, and representing large integers with only boolean variables, is quite a demanding workload even for a computer.

Another great finding lies within the consistency of each solver over multiple test runs, as described as metric TM3 in Table 5.3. The ILP-Solver calculates the optimal result consistently over all test runs with no apparent time deviations. Even though the SAT-Solver most of the time calculates consistently optimized results within a minimal range of cost as well, there are some cases in which the SAT-Solver computes a broader range of objectives. The evaluation of the SAT-Solver on this subject is therefore more decisive. The SAT-Solver often does not find the most optimal solution, but it is interesting to investigate how the results change over each run. The following Figure 5.2 presents one meaningful example. The visualization represents the data collected in the test assessments for the course scheduling problem using the medium complexity group and the limitation of 15 minutes. Here, the lower bound, median, and upper bound are compared across all test runs in order to derive a more reliable conclusion. The objective costs are displayed in ratio to each other, with the highest quality solution receiving the value 1.

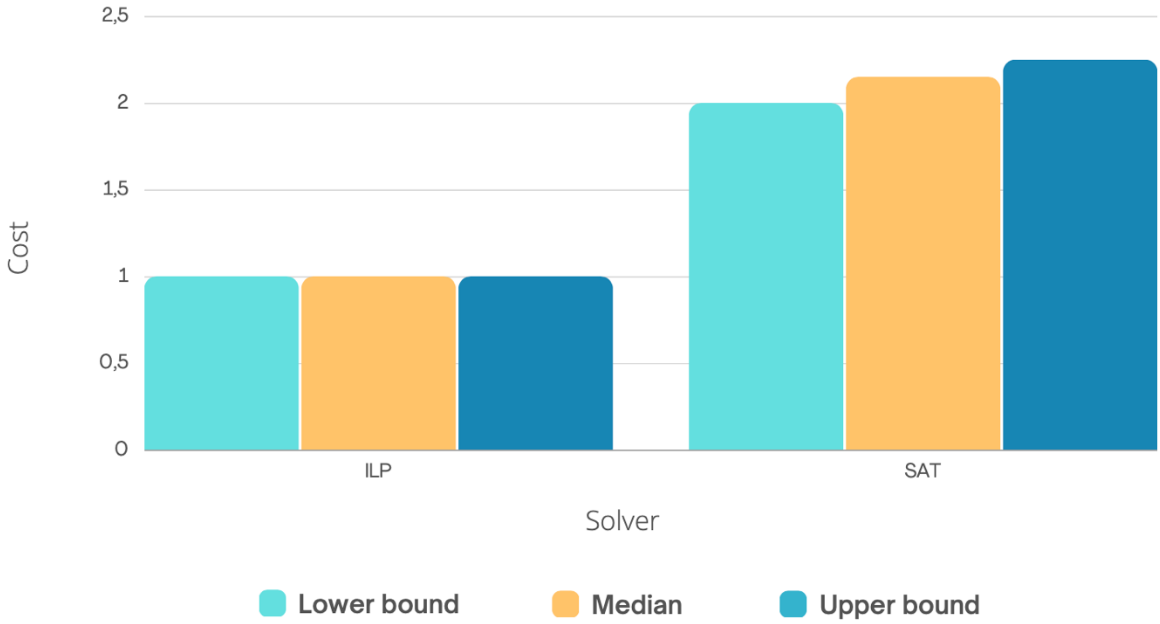

Figure 5.2: Comparison of the consistent performance for both the ILP-Solver and SAT-Solver

The illustration clearly shows the consistent computation of the optimal solution over all three test runs. Whereas the SAT-Solver produces a median with an approximately 2.2 times higher objective. The lower and upper bounds of the SAT-Solver range from 2 times the cost to approximately 2.3 times the cost of the ILP-Solver. The Figure displays proof that the ILP-Solver does outperform the SAT-Solver when it comes to the consistency. An additional impressive remark is, that whereas the SAT-Solver required the full 15 minutes to calculate the results for each run, the ILP-Solver only needed around 10 seconds per run to come up with the optimal solution.

Another trend that we observed is the high memory consumption for the SAT-Solver compared to the ILP-Solver. This is is especially visible for big complexity class problems as can be seen in Figure 5.3.

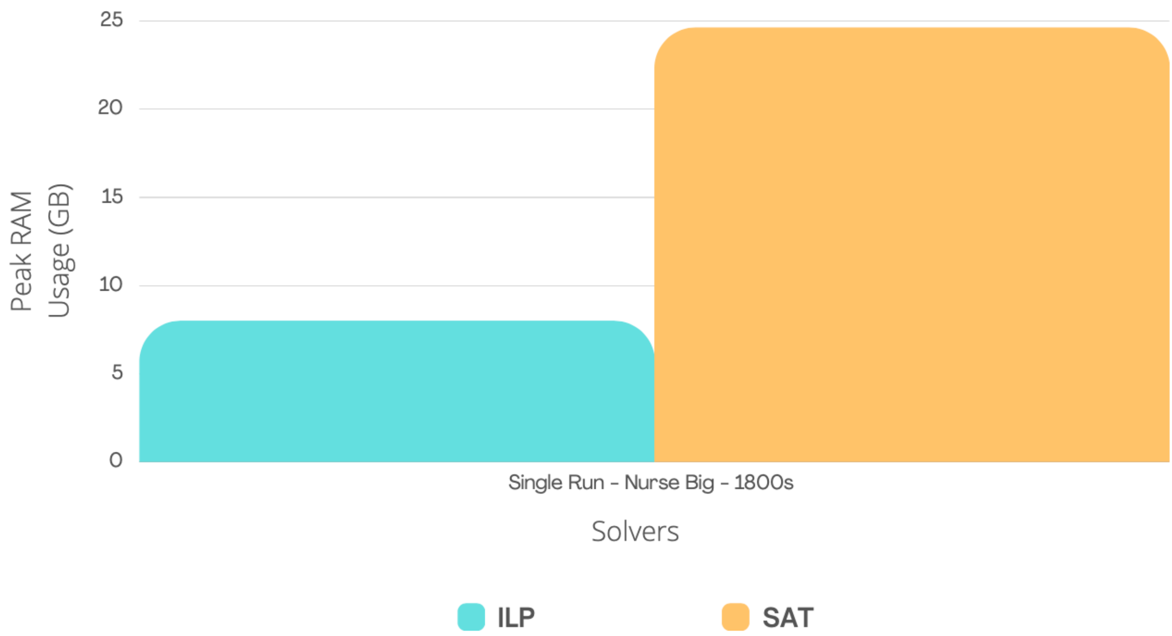

Figure 5.3 Comparision of the peak RAM usage (GB) for both solvers

This could be attributed to implementation problems or environmental issues though and is therefore not relevant as a test metric, but it is interesting to see nonetheless.

In summary, all the test data examined indicate that the ILP solver is more performant than the SAT solver. At least under the implementation of our universal models. However, it must be stated that the model was primarily implemented for the ILP solver in this thesis. The SAT solver copies this implementation and is therefore not specifically tweaked and optimized, since the exploration of the optimal performance is not the goal of this thesis. Finally, we can say that the SAT solver is definitely an alternative to the ILP solver. It can also optimize all types of scheduling problems within acceptable computation time. In addition, another advantage is that it can be used free of charge.

6 Conclusion

Previous chapters show that the prospect of a universal model for scheduling problems is not too farfetched. By defining the four base parameters as people, time slots, tasks, and locations, we demonstrate the formulation of a wide range of constraints over the base parameters. Regrettably, a certain amount of expertise in mathematical modeling continues to be required in order to comprehend and adopt the proposed model. By applying it to three scheduling problems of various types and domains, we demonstrated the models' applicability to real-world scenarios. The three types of scheduling problems are nurse scheduling, course scheduling, and examination scheduling. The universal model already incorporates a vast range of soft and hard constraints to satisfy all the demands of these three problems. However, other types of scheduling problems could be additionally analyzed and more constraints can be formulated to even further generalize the universal model. Thus, the proposed model satisfies the essential question of this thesis.

A secondary goal of this paper is the assessment of a SAT-Solver as a viable alternative to the ILP-Solver. To this end, the proposed model was implemented as a proof of concept for both solvers and thoroughly tested. Evaluating the results led to decisive observations. Overall, the SAT-Solver proves to be a decent alternative to the ILP-Solver for small and medium-sized problems, but with increasing scenario complexity, the free-to-use solver struggles to keep up with the expensive gurobi optimizer. However, the proposed model was primarily developed for the gurobi ILP solver and the SAT solver mirrors the implementation almost exactly. Therefore, the performance of the SAT solver can certainly be further improved. Additionally, it may be interesting to further test the scalability of the ILP-Solver with even larger data sets and time limitations to further investigate the upper bound of the solver.

One last purpose of this study is the development of a web application, which allows the creation of custom scheduling problems using the implementation of our proposed model. The custom problem can afterward be optimized using either the gurobi ILP-Solver or the CaDiCal SAT-Solver. The results are then displayed in form of a timetable on the web application.

In summary, we recommend the continued research into the usage of SAT-Solvers to solve scheduling problems, as it can be a cost-efficient alternative to ILP-Solvers in some of the circumstances described. We hope to have shown enough points of interest that constitute such continued research and thereby consider our goals met.

List of references

[1] A. Wren, “Scheduling, timetabling and rostering — A special relationship?,” in *Practice and Theory of Automated Timetabling*, Berlin, Heidelberg, 1996, pp. 46–75. doi: 10.1007/3-540-61794-9_51.
[2] M. Soares Ferreira da Silva Rocha, “The staff scheduling problem: a general model and applications,” Faculdade de Engenharia da Universidade do Porto, 2013.
[3] N. a. H. Aizam and V. Uvaraja, “Generic Model for Timetabling Problems by Integer Linear Programming Approach,” *Int. J. Math. Comput. Sci.*, vol. 9, no. 12, pp. 718–725, Oct. 2015.
[4] P. D. Causmaecker, P. Demeester, G. V. Berghe, and B. Verbeke, “Analysis of real-world personnel scheduling problems,” p. 15.
[5] J. Gnos and M. Hammel, “Optimierung von Arbeitsplätzen im Gesundheitswesen,” Fachhochschule Nordwestschweiz, Hochschule für Technik, Windisch, 2021.
[6] B. Cheang, H. Li, A. Lim, and B. Rodrigues, “Nurse rostering problems––a bibliographic survey,” *Eur. J. Oper. Res.*, vol. 151, no. 3, pp. 447–460, Dec. 2003, doi: 10.1016/S0377-2217(03)00021-3.
[7] C. Eggenschwiler and P. Jüttner, “Optimal Scheduling of Project Presentations,” Fachhochschule Nordwestschweiz, Hochschule für Technik.
[8] Felix, Simon, “SATInterace.” [Online]. Available: https://github.com/deiruch/SATInterface

Declaration of academic honesty

«We the undersigned declare that all material presented in this bachelor thesis is our own work and written independently only using the indicated sources. The passages taken verbatim or in content from the listed sources are marked as a quotation or paraphrased. We declare that all statements and information contained herein are true, correct and accurate to the best of our knowledge and belief. This paper or part of it have not been published to date. It has thus not been made available to other interested parties or examination boards.»

Windisch, 19.08.2022

Name: Elias Mjoen

Signature:

Name: Joshua Schärer

Signature:

Appendix

A All tests conducted for the nurse scheduling problem

TS1	Nurse scheduling (INF = infeasible, OOM = Out of Memory, TL = Time Limit)							
Comlexity	Limitation	Solver	Iteration 1		Iteration 2		Iteration 3	
			Objective	Solving time (s)	Objective	Solving time (s)	Objective	Solving time (s)
Small	10s	ILP	24	0.264	24	0.527	24	0.453
		SAT	96	TL	93	TL	44	TL
	30s	ILP	24	0.495	24	0.545	24	0.203
		SAT	53	TL	40	TL	47	TL
	1m	ILP	24	0.309	24	0.317	24	0.446
		SAT	26	TL	24	TL	31	TL
Medium	5m	ILP	581	TL	581	TL	581	TL
		SAT	771	TL	794	TL	775	TL
	15m	ILP	581	567.191	581	564.507	581	676.301
		SAT	788	915.658	790	916.023	799	922.18
Big	30m	ILP	1122	TL	1122	TL	1122	TL
		SAT	1229	TL	1236	TL	1229	TL
	1h	ILP	1122	TL	1122	TL	1122	TL
		SAT	1227	TL	1235	TL	1230	TL

B All tests conducted for the course scheduling problem

TS2	Course scheduling (INF = infeasible, OOM = Out of Memory, TL = Time Limit)							
Comlexity	Limitation	Solver	Iteration 1		Iteration 2		Iteration 3	
			Objective	Solving time (s)	Objective	Solving time (s)	Objective	Solving time (s)
Small	10s	ILP	12917	0.111	12917	0.05	12917	0.116
		SAT	17886	TL	16447	TL	17375	TL
	30s	ILP	12917	0.110	12917	0.060	12917	0.057
		SAT	15661	TL	15530	TL	15196	TL
	1m	ILP	12917	0.110	12917	0.058	12917	0.058
		SAT	14563	TL	14906	TL	14770	TL
Medium	5m	ILP	78206	11.191	78206	11.237	78206	10.956
		SAT	INF	TL	INF	TL	INF	TL
	15m	ILP	78206	11.132	78206	10.895	78206	10.907
		SAT	167554	TL	174931	TL	158279	TL
Big	30m	ILP	135022	324.554	135025	357.792	135024	349.868
		SAT	INF	TL	INF	TL	INF	TL
	1h	ILP	135023	318.773	135021	316.571	135021	306.824
		SAT	INF	TL	INF	TL	INF	TL

C All tests conducted for the examination scheduling problem

TS3	Examination scheduling (INF = infeasible, OOM = Out of Memory, TL = Time Limit)							
Comlexity	Limitation	Solver	Iteration 1		Iteration 2		Iteration 3	
			Objective	Solving time (s)	Objective	Solving time (s)	Objective	Solving time (s)
Small	10s	ILP	29	0.077	29	0.035	29	0.035
		SAT	29	0.675	29	0.584	29	0.593
	30s	ILP	29	0.075	29	0.034	29	0.035
		SAT	29	0.674	29	0.592	29	0.581
	1m	ILP	29	0.077	29	0.034	29	0.036
		SAT	29	0.681	29	0.587	29	0.575
Medium	5m	ILP	219	22.813	219	23.745	219	27.131
		SAT	229	TL	229	TL	229	TL
	15m	ILP	219	26.054	219	24.254	219	24.089
		SAT	229	TL	229	TL	229	TL
Big	30m	ILP	421	567.745	421	1425.315	421	1151.24
		SAT	428	TL	428	TL	428	TL
	1h	ILP	421	1350.578	421	995.768	421	634.98
		SAT	428	TL	428	TL	428	TL