# Scheduling Physicians using Constraint Programming 

## Samuel Edqvist

UPPSALA UNIVERSITET

## Teknisk- naturvetenskaplig fakultet

 UTH-enheten
## Besöksadress:

Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

## Postadress:

Box 536
75121 Uppsala

## Telefon:

018-47130 03
Telefax:
018-4713000

## Hemsida:

http://www.teknat.uu.se/student

## Abstract <br> Scheduling Physicians using Constraint Programming

Samuel Edqvist

The thesis demonstrates the suitability, flexibility and efficiency of applying constraint programming for an automatic generation of schedules for physicians. Constraint programming is about assigning values to a number of variables subject to constraints. In personnel scheduling, a variable represents a staff member at a certain time and these variables are assigned work tasks or shifts. Examples of constraints are that a limited number of doctors can do operation simultaneously and that there is a lowest number of appointments that must be scheduled each week. A real life problem from a Swedish clinic has been studied and a prototype software, without any optimisation, has been developed. Tests with random pre-assignments have been carried out for different variable orders. The clinic's constraints were formulated as two global constraints: count and logical constraints. The software successfully makes schedules for some instances, but to make it more dynamic, real life pre-assignments and constraints on individual doctors should be tested interactively, using an appropriate user interface.

[^0]
## Sammanfattning

Schemaläggning av personal är ofta en mycket tidskrävande och svår process. Det finns dels generella krav som gäller verksamheten i stort och dels krav som gäller varje individ, enligt lagstiftning, fackliga regleringar och personliga önskemål. Alla dessa krav ska uppfyllas samtidigt.

Inom en verksamhet som vård är det speciellt svårt eftersom vården ska vara tillgänglig 24 timmar om dygnet, 7 dagar i veckan. Läkarna på en klinik är en begränsad resurs som man vill få ut maximalt av samtidigt som man uppfyller de individuella önskemål som finns i största möjliga mån. Dessutom måste man ta hänsyn till läkarnas olika erfarenhet och specialistkompetenser. Verksamheten är oftast så komplex att det är en läkare själv som måste utföra schemaläggningen, vilket också tar mycket av dennes arbetstid. Trots att schemaläggning är ett välstuderat problem, är det fortfarande brist på bra programvara i Sverige, som effektivt kan utföra automatiskt schemaläggning av läkarpersonal.

Ett personalschema består oftast av två axlar: personal och dag. Man får då ett rutnät där uppgiften blir att ge varje person en arbetsuppgift per dag, dvs fylla i ett tecken eller en text i varje ruta. Ofta kan också dagen vara indelad i flera tidsintervall eller moduler med olika tecken. Figuren visar ett exempelschema med fem läkare.
Här används fem olika tecken:"J" - Jour, "O" - Operation, "M" - Mottagning, "X" - Ledighet samt "-" - Övrig frånvaro, såsom kurser och uppdrag på annan klinik.

Antag att kliniken har en akutmottagning. För detta krävs att det varje dag, inklusive helger, finns en läkare som går jour. För övrigt krävs det att kliniken har minst 4 mottagningsmoduler och 5 operationsmoduler i veckan. Det finns endast två operationssalar, därför kan maximalt två läkare ha operation en och samma dag. Antag att en mottagningsdag är påfrestande och därför gäller det att en läkare ska ha maximalt en mottagningsdag per vecka. Vidare måste läkare som har jour vara lediga dagen efter. Detta kan sammanfattas med följande villkor:
Antalet J per dag = 1,
Antalet O per dag $\leq 2$,
Antalet M per vecka $\geq 4$,
Antalet O per vecka $\geq 5$,
Antalet M per läkare $\leq 1$,
$\mathrm{J} \Rightarrow \mathrm{X}$ dagen efter.
Verifiera gärna själv att alla dessa villkor uppfylls i exempelschemat ovan. Om vi nu ska låta ett datorprogram automatiskt generera ett liknande schema som uppfyller villkoren måste vissa inmatningar ske först. Eftersom kurser och uppdrag på annan klinik ("-") är sådant som bestäms någon annanstans, så måste dessa fyllas i manuellt, innan schemat genereras. Nu var detta exempelschema ganska litet och består endast av 35 rutor. I verkliga fall kan antalet läkare vara 20, dagen delas upp i tre moduler och schemaläggningshorisonten 2 månader, vilket ger ungefär 3600 rutor som ska fyllas i. Något som kan bli mycket rörigt och tidskrävande när det

| Läkare | Måndag | Tisdag | Onsdag | Torsdag | Fredag | Lördag | Söndag |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Dr. A | J | X | O | - | O | X | X |
| Dr. B | M | J | X | O | O | X | J |
| Dr. C | O | - | J | X | M | J | X |
| Dr. D | M | O | - | J | X | X | X |
| Dr. E | O | - | O | M | J | X | X |

Exempelschema
görs manuellt.
Som exemplet visar så består ett schemaläggningsproblem alltså i att uppfylla ett antal (matematiska) villkor. Det existerar många olika typer av lösningsmetoder för schemaläggning. En metod som har visat sig bra är villkorsprogrammering (constraint programming). Det som är speciellt med villkorsprogrammering är att man skiljer på villkoren som definierar problemet och algoritmerna som ska hitta en kombination som uppfyller villkoren. Förutom att schemaläggningsmjukvara som bygger på villkorsprogrammering snabbt kan hitta lösningar, så är det också ganska naturligt att formulera problemet med villkor.

Ett problem som uppstår är dock när alla de villkor man ställt upp inte tillsammans kan uppfyllas. Om vi återgår till exempelschemat och tänker oss att Dr. B istället är frånvarande på Måndagen och att Dr. A inte får ha mottagning överhuvudtaget. Därmed finns bara 3 mottagningsmoduler i schemat och villkoret var minst 4. För att korrigera detta låter vi Dr. C istället ha mottagning på måndagen. Detta medför att Dr. C har 2 mottagningsmoduler istället för 1 som var maxvillkoret, men det föregående villkoret är viktigare och båda kan inte uppfyllas samtidigt. Det första villkoret kallas därför för ett hårt villkor, det måste alltid var uppfyllt, medan det andra kallas mjukt villkor, det är önskvärt att det uppfylls, men man får bryta mot det, om nödvändigt. Målet, när mjuka villkor är inblandat, är att uppfylla så många mjuka villkor som möjligt och samtidigt uppfylla samtliga hårda. Det finns också metoder som hanterar att de mjuka villkoren prioriteras olika.

I det här arbetet har ett specifikt problem från ortopedkliniken på Kalmar sjukhus studerats och en prototypmjukvara för att generera scheman åt kliniken har utvecklats. Syftet är att demonstrera hur effektivt villkorsprogrammering är för schemaläggning och att ge en bild av hur schemamodellen för läkare kan se ut i Sverige. Klinikens villkor visade sig vara begränsade till "antaletvillkor" (liknande de sex första villkoren i exemplet) och logiska villkor (liknande det sistnämnda i exemplet). Villkoren har anpassats så att schemaläggning enbart blir ett beslutsproblem, dvs. inga mjuka villkor används. Programmet kan generera scheman, men det förutsätter att det finns förifyllda tecken ("-" i exempelschemat) och dessa måste ges av kliniken. Slumpmässigt angivna förifyllda värden har använts för att göra tester på olika metoder som söker efter lösningar.

## Contents

1 Introduction ..... 3
1.1 Background ..... 3
1.2 Previous Work ..... 3
2 Theory ..... 5
2.1 Constraint Programming ..... 5
2.1.1 Constraint Satisfaction ..... 5
2.1.2 Searching ..... 6
2.1.3 Optimisation ..... 7
2.2 Global Constraints ..... 7
2.2.1 Count ..... 7
2.2.2 Sum ..... 8
2.2.3 Reification ..... 8
3 Methodology ..... 9
3.1 Schedule Model ..... 9
3.2 The Kalmar Model ..... 10
3.2.1 Background ..... 10
3.2.2 Scheduler Implementation ..... 10
3.2.3 Decision Variables and Constraints ..... 12
3.3 Used Software ..... 16
4 Experiments and Results ..... 17
4.1 Description ..... 17
4.2 Testing System ..... 17
4.3 Variable Orders ..... 18
4.4 Results ..... 18
5 Conclusions ..... 22
5.1 Restrictions and Delimitations ..... 22
5.2 Future Directions ..... 23
Acknowledgements ..... 24

References 24

## Chapter 1

## Introduction

### 1.1 Background

Scheduling of any staff can be both hard, time-consuming and tedious. In health care, work must be carried out at a continuous basis, 24 hours a day, 7 days a week. What is particularly difficult when it comes to doctor's rostering is that the access is very limited and that the doctors' level of experience and their special competence varies even within one clinic. Moreover, the activity of making schedules for doctors can also be quite complex which is why the scheduler is normally a doctor them self, whose valuable time can be used more effectively, giving patients medical assistance instead.

Scheduling in health care is a well studied and difficult task. Most of the time it is still solved by hand in a lengthy process and where automated systems are involved they tend to be strongly linked to a particular context [1]. Scheduling problems are usually defined by time and capacity constraints which can be formulated in an easy way [2] and constraint programming has shown to be very efficient for short-term scheduling.

The department of orthopaedics, Kalmar hospital, Sweden experiences that their scheduling takes up a lot of time and energy from the scheduler particularly and that an efficient software that can automatically generate schedules would be of much help. They have not found any such efficient software on the Swedish market, however. This is the case for several clinics in Sweden.

In this thesis I am concentrated on making a model for the Kalmar clinic specifically. This will hopefully give a good start for future research towards a more generic model. No optimisation techniques are involved, the problem is treated as a decision problem.

### 1.2 Previous Work

[1] recognises two families of staff scheduling: rotating schedules and personalised schedules. Rotating schedules give a repeating pattern of sequences of work and
rest days alternating over a certain period. The individual schedules are then identical but out of phase with each other, ensuring a fair schedule (see [3]). This approach is only appropriate when the staff is interchangeable. Very often in physician scheduling however this is not the case since work differs due to things as seniority, unavailabilities due to other activities such as courses and special preferences.

The "HIBISCUS" software in [1] uses a model with different types of global constraints that captures a lot of different rules. HIBISCUS have been successfully applied both to a nurse scheduling instance as well as three different physician instances, all in the Montréal area.

Internationally there exists a few commercial software that are more or less specialized on physician scheduling. "Physician Scheduler" by AtStaff purely specialises in physician scheduling for different types of clinics and Omega Optimisation's products have been successfully applied on a hospital in Quebec. In Sweden there exists software that deals with scheduling of physicians that guarantees no double bookings and that gives valuable statistics that help the scheduler to make it fair. None of those have fully broken through, however.

## Chapter 2

## Theory

### 2.1 Constraint Programming

Constraint programming is used to solve combinatorial problems by stating constraints on and between variables. This can make it particularly powerful in applications where it is natural to describe the problem with constraints. Such applications are, beside from scheduling: finance, logistics, transport etc. The power of constraint programming is that it separates the definition of the constraints that define the problem and the algorithms and heuristics for solving them. If the constraints can be expressed or broken down to some more generic constraints, called global constraints (see 2.2), the algorithms can be gotten "for free", if available. Efficient such algorithms are available in different constraint programming libraries for many different global constraints and new ones can be developed. In this section, the most relevant parts of constraint programming are explained.

### 2.1.1 Constraint Satisfaction

A constraint satisfaction problem (CSP) consists of (discrete) variables, called decision variables, and a number of constraints that acts on these variables. The decision variables are associated with a domain, that is a non-empty set, specifying what values they can take. The goal with the CSP is to assign values to all variables within their respective domain, so that all constraints are satisfied.

A typical example of a CSP is the game Sudoku. Sudoku consists of a $9 \times 9$ grid where each square is either empty or has a digit between 1 and 9 . The goal is to fill in digits between 1 and 9 in the empty squares, so that: all the squares in each row, column and $3 \times 3$ sub square, takes different values respectively. Here the decision variables are the squares and the domain is (initially) $[1,9]$ for all the empty squares and the others are assigned to their pre-filled value.
Let X denote a matrix with all the squares. Then the constraints are:
alldifferent $\left(X_{i, 0}, \ldots, X_{i, 8}\right) \mathrm{i}=0, \ldots, 8$ (rows)
alldifferent $\left(X_{0, j}, \ldots, X_{8, j}\right) \mathrm{j}=0, \ldots, 8$ (columns)
alldifferent $\left(X_{3 \cdot\lfloor i / 3\rfloor+\lfloor j / 3\rfloor, 3 \cdot i \bmod 3+j \bmod 3}, \mathrm{j}=0, \ldots, 8\right) \mathrm{i}=0, \ldots, 8$ (sub squares) where alldifferent(A) guarantees that all variables in A takes different values. Note that 27 "alldifferents" are posted here, since there are 9 columns, rows and subsquares, respectively.

### 2.1.2 Searching

When a CSP has been defined, that is, all constraints have been posted, the solving begins. The first step is the so called constraint propagation that is supposed to use the constraints to prune values from the domains. Using alldifferent for example on $\langle\{1,2,3\},\{1,2,4\}, 1\rangle$, can be reduced to $\langle\{2,3\},\{2,4\}, 1\rangle$, removing 1 from the first two domains as it is already assigned to the third variable.

In the example above, there are still several possible combinations left, but not all satisfies alldifferent, which is why another technique on top of the constraint propagation must be introduced, namely branching. Branching is a way of testing by assigning certain variables certain values or just decrease their domain by removing values. After a branch the constraint propagation is carried out again, and if there are still more than one value, in one or more domains, the branching continues. In the previous example lets say we try by assigning the first variable to 2 , then propagation will remove 2 from the second variable and we have one single combination left; $\langle 2,4,1\rangle$ which satisfies alldifferent. This is one solution, if we want to find more solutions we back track to $\langle\{2,3\},\{2,4\}, 1\rangle$ and try with other values. Back tracking is also carried out when branching leads to a combination that fails to satisfy a constraint. Branching is illustrated by a binary search tree, as in figure 2.1.


Figure 2.1: A binary search tree. The diamonds are solutions, the squares are where the branching got to a fail and the circles are where there exists unassigned decision variables after propagation.

Branching uses a certain heuristic for in what order to pick variables to branch on, the variable order, and between what values in that variable's domain, the branching should take place, the value order. Common variable orders are largest domain size first or the one with the smallest minimum value, largest maximum value, smallest maximum value etc. The value order can be to first assign to the smallest or to the largest value or to split the domain in half and remove either the upper or the lower half. Picking the right variable and value order is crucial for performance and it is very problem dependent. For this reason it is important to test many different heuristics empirically. A good strategy however is to pick a heuristic that "fails first".

### 2.1.3 Optimisation

Optimisation is about minimising or maximising an objective function, sometimes subject to a number of constraints. By adding some objective function of the decision variables in a CSP, we have a constraint optimisation problem. Constraint optimisation can have two types of constraints: hard and soft. All the hard constraints must be satisfied, but the soft constraints can be violated. The optimising criterion is then to maximise the number of satisfied soft constraints, possibly weighed differently, subject to all of the hard constraints. The use of soft constraints is common in scheduling problems since there is often rules, for example concerning comfort, that can't always be satisfied simultaneously as all the hard ones.

### 2.2 Global Constraints

A global constraint is a generic constraint. One that was mentioned earlier is "alldifferent" that guarantees that all decision variables it acts on are assigned different values respectively. "Sum" is another that constrains the sum of some variables to be in some relation to a value. There are good propagation algorithms, known as filtering algorithms, available for many global constraints in different CP libraries, including Gecode. In the rest of this section the global constraints that are used in the Kalmar model are explained . ${ }^{1}$

### 2.2.1 Count

Consider a vector of integer decision variables $V$, two constants $c_{1}$ and $c_{2}$ and a relation operator $R$.
$\#\left\{V=c_{1}\right\} \mathrm{R} c_{2}$
Guarantees that the number of variables in V that takes the value $c_{1}$ is in relation R to $c_{2}$. If R is $=, c_{2}$ can also be a decision variable.

Example: R is $=, c_{2}=\{1,4,5\}$ and $c_{1}=0$

[^1]$\#\{V=0\}=\{1,4,5\}(\#\{V=0\} \in\{1,4,5\})$
Guarantees that the number of variables in $V$ that takes the value 0 is either 1,4 or 5. The constraint has arc-consistent filtering [4].

### 2.2.2 Sum

Consider a vector of integer decision variables $V$, a constant c and a relation operator R.

$$
\sum V \mathrm{Rc}
$$

Guarantees that the sum of variables in V is in relation R to c . If R is $=$, then c can also be a decision variable.

### 2.2.3 Reification

Consider a condition c and a boolean decision variable b. c can be reified into b denoted
$\mathrm{c} \Leftrightarrow \mathrm{b}$
Constraints can then be posted on two or more of these booleans, particularly logical constraints between two booleans can be useful.
Example: Consider two decision variables $x$ and $y$, let
$\mathrm{x}<5 \Leftrightarrow \mathrm{~b} 1$
$\mathrm{y}>4 \Leftrightarrow \mathrm{~b} 2$
then
$\mathrm{b} 1 \Rightarrow \mathrm{~b} 2$
will post the constraint $\mathrm{y}>4$ if $\mathrm{x}<5$.
Reification should generally be avoided since it introduces many propagators which yields weak propagation.

## Chapter 3

## Methodology

### 3.1 Schedule Model

A schedule normally consists of variables in two dimensions: the time and the staff, divided into teams or individuals. In this case, the time index can represent a single day and the value that the variables are assigned, would correspond to a work task in combination with a certain shift (time slot), for each staff member respectively. Instead of combining a work task with a certain time slot, the schedule can be extended with a third dimension representing the time slots. The time slots are hereafter called modules. The schedule is then a 3D-matrix where a variable represents what a certain doctor does at a certain day in a certain module, this is hereafter called a cell. There is an important trade off here: using the 2D-schedule decreases the number of variables, but the number of possible values can become very large. The number of possible permutations of assignments is
$T^{M}$
where $T$ is the number of possible tasks and $M$ is the number of cells. Most of these permutations can of course be ruled out by constraints stating that not all tasks can be assigned to any single cell independently, some tasks are for example, whole days only. This will implement such constraints directly in the model instead which is often much better. One disadvantage with this approach is that if the clinic wants to change constraints, the whole model might need to be changed.

In nurse scheduling and often also for doctors the sense of different work tasks is not visible in the schedule(see for example [5] and [6]) but only what shift (e.g. morning, day or night) a certain staff member is assigned at a certain day. In this case, day variables is the obvious choice.

The main categories of rules from hospitals in the Montreal area and other relevant literature is presented in [1]. Based on this and my own experience from the Kalmar case, here are the categories:

Possible Combinations of Modules for Tasks. Even though there are several modules per day, some tasks are not allowed to be scheduled to any combination of modules for a single day and doctor. Some tasks are for example whole days only.

Demand. A sufficient number of cells of the different work tasks must be staffed throughout every day, week, month or the whole scheduling horizon to guarantee minimum coverage.

Staff Availability. A given staff member are not available for all tasks based on his/her qualification, therefore assignments to such tasks for that staff member must be forbidden. A staff member can also be unavailable at a certain cell because of vacation or other leisure, part time status or because of outside responsibilities, courses etc. These are pre-assigned.

Room Availability. There is a limited amount of wards, surgery rooms, special instruments etc. at certain times.

Distribution. Scheduling should guarantee a somewhat fair distribution of different tasks among the staff. This is about balancing a certain type of shift among the staff, either evenly or according to some criterion such as seniority and to give each doctor an individual schedule with an even distribution of the different tasks. Also the number of weekends off should be well distributed.

Ergonomics. The individual schedules should give the doctors continuity in some tasks and not working too many heavy tasks in the same week or month.

The rules regarding demand and staff/room availability are always hard. Some of the distribution and ergonomic rules must sometimes be relaxed, or treated as soft.

### 3.2 The Kalmar Model

### 3.2.1 Background

The clinic has 19 doctors and schedules are made with 8 weeks horizon. Work hours are fixed every day for every doctor except that two doctors are on-call the whole day, partially at the hospital and partially at home. One of the on-calls is primary and the other one is back-up. The primary on-call is scheduled only as such the whole day, the back-up on-call however is also scheduled as simultaneous with other tasks. Every day is divided into 3 modules that correspond to three time intervals of approximately the same length. This system is chosen for simplification and to give a decent overview and does not always exactly correspond to how they actually work. These cells are assigned one out of several different work task or leisure. There are differences in the doctors' seniority as well as special training which affects availability as mentioned in 3.1.

Figure 3.1 shows a sample schedule from Kalmar constructed by the clinic's scheduling person. See 3.2.2 for explanation of the different characters, everything that is not listed there is treated as miscellaneous.

### 3.2.2 Scheduler Implementation

Most of the assignments have precise constraints regarding them and can therefore be assigned by the scheduling software. Other assignments are of such nature that


Figure 3.1: Real man-made schedule from the Kalmar clinic.
they are rather done manually. Assignments of the latter type can be either assigned before search (pre-assigned) or after search (post-refined).

The constraints are spanning over one month at most and searches are therefore made with that horizon, even though the actual planning horizon is larger.

Two schedules must be searched for since back-up on-call is, on week days, done simultaneously with other tasks. The main schedule is represented with a 3d-matrix X , where $\mathrm{X}_{i, j, k}$ states what doctor i is doing at day j in module k . The back-up on-call schedule is represented with a vector Xb where $\mathrm{Xb}_{j}$ represents who that has back-up on call at day j. Exactly one doctor is back-up on-call every day. The doctors are divided into two groups: chief physicians (Cph) and non-chief physicians (Uph).

This implementation uses 6 different values in the main schedule and 11 values in the back-up on-call (the total number of Cph doctors). One of the assignments in the main schedule is "miscellaneous", representing tasks that are only assigned manually, either before or after search. If the value miscellaneous is pre-assigned, it means that the cell is assigned a task that is not represented by the other values. If it is assigned by the scheduling software, it means that the cell is available for any task and this must be specified manually.
Let
$\mathrm{n}=$ "the total number of staff members"
$\mathrm{d}=$ "the scheduling horizon"
$\mathrm{m}=$ "the number of modules per day"
then the number of variables(cells) in this model is $n \cdot d \cdot m+d=19 \cdot 28 \cdot 3+28=1624$
where h is added for the back-up on-call schedule. This is a quite large instance.
The different assignments in the main schedule are:
P - Primary on-call
W - Ward
A - Appointments
O - Operation
x - Off

-     - Pre-assigned miscellaneous task (value NA)

〈blank〉 - Cell is available. (same value as "-", never pre-assigned)
The value for $\langle$ blank $\rangle$ and "-" will be called NA. The back-up on-call is marked with a thick red border around the cells corresponding to the day and doctor.

### 3.2.3 Decision Variables and Constraints

3d-matrix $X$ and vector $X b$ of decision variables.
$X_{i, j, k}=$ doctor $i$ 's task at day $j$ in module $k$
$X_{i, j, *}$ is all decision variables for doctor $i$ at day $j$ etc.
$X b_{j}=$ which doctor goes back-up on-call at day $j$.
There are n doctors, d days and m modules. Furthermore, let $W D=\{$ "All week days (Monday-Friday)" $\}$

```
WD
WeD ={"All week-end days (Saturdays and Sundays)"}
WeD}\mp@subsup{D}{w}{}={"\mathrm{ "All week-end days in week w" }
FtS S = {"Friday to Sunday week w"}
Us={"All Uph doctors"}
Cs={"All Cph doctors"}
NoVac ={"All Doctors with at most 2 weeks on vacation" }
b1, b2,... are booleans.
```


## Generic

g1 Only on-calls and off at week ends. Remove all other values from their domains.
$X_{i, j, k} \neq\{\mathrm{A}, \mathrm{W}, \mathrm{O}, \mathrm{NA}\}, \forall i, k$ and $j \in W e D$
g2 Offs are whole days only. Either 0 modules (not off) or m modules (off all modules).

$$
\#\left\{X_{i, j, *}=\mathrm{x}\right\} \in\{0, \mathrm{~m}\}, \forall i, j
$$

## Back-up On-call (B)

B1 Whole days only.
Treated by the model, since the back-up on-call schedule only has one variable per day.

B2 Only Cph doctors.
Treated by the model, since the back-up on-call schedule only has the Cph doctor's indices.

B3 One doctor every day.
Treated by the model.
B4 Cannot be B when off.
$\mathrm{b} 1 \Leftrightarrow\left(X_{i, j, 0}=\mathrm{x}\right), \mathrm{b} 2 \Leftrightarrow\left(X b_{j}=\mathrm{i}\right), \neg(\mathrm{b} 1 \wedge \mathrm{~b} 2), \forall j, \forall i \in C s$
Presupposes that $\mathbf{g} 2$ is posted.
B5 Week end B is always Friday, Saturday and Sunday.
$\#_{j \in F t S_{w}}\left\{X b_{j}=\mathrm{i}\right\} \in\{0,3\}, \forall i, w$
B6 Always off the Friday after a week end B.
$\mathrm{b} 1 \Leftrightarrow\left(X b_{j}=\mathrm{i}\right), \mathrm{b} 1 \Leftrightarrow\left(X_{i, j+5,0}=\mathrm{x}\right), \forall j \in$ Sundays $\mid j \leq \mathrm{d}-1, \forall i \in C s$ Presupposes that $\mathbf{g 2}$ is posted.

B7 At most one week day B per week.
$\#_{j \in W e D_{w}}\left\{X b_{j}=i\right\} \leq 1, \forall w, \forall i \in C s$
B8 At most 4 B per month.
$\#\left\{X b_{*}=i\right\} \leq 4, \forall i \in C s$

## Primary on-call (P)

P1 P is whole day only. Either 0 modules (not P ) or m modules ( P all modules).
$\#\left\{X_{i, j, *}=\mathrm{P}\right\} \in\{0, \mathrm{~m}\}, \forall i, j$
P2 No P for Cph.
$X_{i, j, k} \neq \mathrm{P}, \forall i \in C s, \forall j, k$
P3 One doctor every day.
$\#_{i \in U s}\left\{X_{i, j, 0}=\mathrm{P}\right\}=1, \forall j$
Presupposes that $\mathbf{P} 1$ is posted.
P4 Always off the day after a P.
$\mathrm{b} 1 \Leftrightarrow\left(X_{i, j, 0}=\mathrm{P}\right), \mathrm{b} 2 \Leftrightarrow\left(X_{i, j+1,0}=\mathrm{x}\right), \mathrm{b} 1 \Rightarrow \mathrm{~b} 2, \forall j \leq \mathrm{d}-1, \forall i \in U s$
Presupposes that $\mathbf{P} 1$ and $\mathbf{g 2}$ are posted.
P5 2-4 weekday Ps per month (period of four weeks).
$2 \leq \#_{j \in W e D}\left\{X_{i, j, 0}=\mathrm{P}\right\} \leq 4, \forall i \in U s$
Presupposes that $\mathbf{P} 1$ is posted.
P6 P on Friday is always together with P on Sunday.
$\#\left\{\left\langle X_{i, j, 0}, X_{i, j+2,0}\right\rangle=\mathrm{P}\right\} \in\{0,2\}, \forall j \in$ Saturdays, $\forall i \in U s$
Presupposes that $\mathbf{P} 1$ is posted.
P7 At most one weekend P per month (period of four weeks).
$\#_{j \in W e D}\left\{X_{i, j, 0}=\mathrm{P}\right\} \leq 1, \forall i \in U s$
Presupposes that $\mathbf{P} 1$ is posted.
P8 At most one weekday P per week.
$\#_{j \in W_{e} D_{w}}\left\{X_{i, j, 0}=\mathrm{P}\right\} \leq 1, \forall i \in U s, \forall w$
Presupposes that $\mathbf{P} 1$ is posted.

## Ward (W)

W1 Always module 0 and 1.
$\#\left\{\left\langle X_{i, j, 0}, X_{i, j, 1}\right\rangle=\mathrm{W}\right\} \in\{0,2\}, \forall i, j$
W2 Never module 2.
$X_{i, j, 2} \neq \mathrm{W}, \forall i, j$
W3 2 Uphs every (week)day.
$\#_{i \in U s}\left\{X_{i, j, 0}=\mathrm{W}\right\}=2, \forall j \in W D$
Presupposes that W1 is posted.
W4 Mondays and Thursdays 2 additional Cphs.
$\#_{i \in C s}\left\{X_{i, j, 0}=\mathrm{W}\right\}=2, \forall j \in$ Mondays $\cup$ Thursdays
Presupposes that W1 is posted.

W5 No Cphs except Mondays and Thursdays.
$X_{i, j, k} \neq \mathrm{W}, \forall i \in C s$ and $\forall k, \forall j \in$ Tuesdays $\cup$ Wednesdays $\cup$ Fridays
W6 The same Cph on Monday and Thursday.
$\#\left\{\left\langle X_{i, j, 0}, X_{i, j+3,0}\right\rangle=\mathrm{W}\right\} \in\{0,2\}, \forall i \in C s$ and $j \in$ Mondays
Presupposes that W1 and W4 are posted.
W7 The same two Uph doctors the whole week.
$\#_{j \in W D_{w}}\left\{X_{i, j, 0}=\mathrm{W}\right\} \in\{0,5\}, \forall i \in U s$ and $\forall$ weeks $w$
Presupposes that W1 and W3 are posted.
This constraint is relaxed when testing some of the real instances, since they are over-constrained: Post the constraint that at most 4 doctors have ward at one week and only allow $1,3,4$ or 5 Ws , to not spread it too much, but to make it possible to satisfy constraint W3.
Introduce matrices of auxiliary variables:
$X w$ - Holding the number of W every doctor has each week.
Xob - Booleans for each doctor and week stating whether a doctor i has W (at least ones) week w.

Count W occurrences for every doctor each week and carry in $X w$ :
$\#_{j \in D W_{w}}\left\{X_{i, j, 0}=\mathrm{W}\right\}=X w_{i, w}, \forall i \in U s$ and $\forall w$
Forbid 2 occurrences:
$X w_{i, w} \neq 2, \forall i, w$
Reify at least one W for every week and doctor into Xob:
$X w_{i, w} \geq 1 \Leftrightarrow X o b_{i, w}, \forall i, w$
Post constraint less than or equal to 4 doctors having W for each week: $\sum_{i} X o b_{i, w} \leq 4, \forall w$

## Appointments (A)

A1 8-14 cells per day except Fridays.
$8 \leq \#\left\{X_{*, j, *}=\mathrm{A}\right\} \leq 14, \forall j \in W D \backslash$ Fridays
A2 At least 42 cells per week.
$\#_{j \in W D_{w}}\left\{X_{*, j, *}=\mathrm{A}\right\} \geq 42, \forall w$
A3 At most 4 doctors at the same time (i.e. the same day and module). $\#\left\{X_{*, j, k}=\mathrm{A}\right\} \leq 4, \forall j, k$

A4 Never A on both the last 2 modules.
$\#\left\{\left\langle X_{i, j, 1}, X_{i, j, 2}\right\rangle=\mathrm{A}\right\} \leq 1 \forall i, j$

A5 Never A on the last module Fridays. $X_{i, j, 2} \neq \mathrm{A}, \forall j \in$ Fridays,$\forall i$

## Operation (O)

O1 Before noon surgeries are the first two modules.
$\#\left\{\left\langle X_{i, j, 0}, X_{i, j, 1}\right\rangle=\mathrm{O}\right\} \in\{0,2\}, \forall i, j$
O2 A certain interval for the number of surgeons before noon (module 0 and 1 ), different for different days in the week.
$\#\left\{X_{*, j, k}=\mathrm{O}\right\} \in\{$ "Before noon demand for j's weekday" $\}, \forall j \in W D, \forall k \in$ $\{0,1\}$
Presupposes that $\mathbf{O 1}$ is posted.
O3 A certain interval for the number of surgeons after noon, different for different days.
$\#\left\{X_{*, j, 2}=\mathrm{O}\right\} \in\{$ "After noon demand for j's weekday" $\}, \forall j \in W D$
Presupposes that O1 is posted.
O4 At least 4 operation days for Uphs per month.
$\#\left\{X_{i, j, k}=\mathrm{O}\right\} \geq 4, \forall i \in U s \cup N o V a c, \forall k \in\{0,2\}, \forall j \in W D$
Presupposes that O1 is posted.
05 At least 8 operation before noon for Cphs per month. $\#\left\{X_{i, j, 0}=\mathrm{O}\right\} \geq 8, \forall i \in C s \cup N o V a c, \forall j \in W D$
Presupposes that $\mathbf{O 1}$ is posted.
O6 At least 2 operation afternoon noon for Cphs per month.
$\#\left\{X_{i, j, 3}=\mathrm{O}\right\} \geq 2, \forall i \in C s \cup N o V a c, \forall j \in W D$
Presupposes that O1 is posted.

### 3.3 Used Software

The programming language that was used was Java both for my familiarity with it and for being easy to debug, which saves much of the time spent on developing. Together with Java, Gecode/J was used, which is a Java wrapper for the C++ constraint programming library Gecode.

Gecode is an open, free and very efficient environment for constraint programming. It supports the programming of new filter algorithms as well as decision variable types and branching strategies. Filter algorithms for all global constraints that are used are included in the current version of Gecode/J that is used here (v2.2.0), as well as several standard branching strategies. As of the end of this work it was announced that the Gecode/J support will be discontinued by its current developers, but this implementation is being ported to $\mathrm{C}++$ and Gecode. For more information on Gecode and documentation see [7].

## Chapter 4

## Experiments and Results

### 4.1 Description

Experiments have been carried out to test the performance for a quite large instance as the Kalmar model is. Many things can be altered here, variable order, value order, pre-assignments, horizon and constraints, yielding many combinations. To limit the number of test runs I have fixed the scheduling horizon, the value order and all constraints are used on all the test runs. The scheduling horizon is fixed to 1 month since no constraint is spanning over a longer period than that. The value order is fixed to assigning values in this order: P, W, A, O, NA, x. This corresponds to the most constrained value first. Note that there are no constraints on "NA" which is why "x" (leisure) will never be assigned by branching in the first solution.

Tests have been done for different branching heuristics with random pre-assignments at each run. The randomisation only pre-assigns to complete vacation weeks in a manner that solutions exists. It should be noted that the random pre-assignments does not correspond to real life. Since these specific constraints are used from the Kalmar model, the purpose of these tests are merely to give an idea of what this scheduling software is capable of and to give a guidance to what branching heuristics are the best. Changing the constraints, even though of the same type, and the number of modules, might change things dramatically.

One test was also made on real-life provided instance from Kalmar using, as needed, a relaxed model by relaxing constraint W8.

### 4.2 Testing System

The testing computer has the following properties:

- Processor: Turion $64 \times 2$ Mobile Technology TL-58 (1.9 GHz, 2*512 KB L2 Cache)
- Memory: 2 GB DDR2 RAM
- Operating System: Windows Vista
- Java version: 1.6.0_07
- Gecode/J version: 2.2.0


### 4.3 Variable Orders

The different variable orders tested are:
MAX_MAX - Branches on the variable with the largest maximum value in its domain. Before the systematic testing it seemed to be the generally best one.
REGRET_MAX_MAX - Branches on the variable with the largest max-regret. Max-regret is the difference between the largest and the second-largest value still in the domain. Before systematic testing it seemed to often be faster than MAX_MAX but with fewer runs that finished in less than 20 min .

These results motivated testing a combination of the heuristics above that starts with REGRET_MAX_MAX stops it after 10 seconds if no solution is found before that. Then a new search using MAX_MAX starts, that runs until a solution is found.
Figure 4.1 shows a generated schedule with random pre-assignments using MAX_MAX.

### 4.4 Results

Using the combination of variable orders, 500 runs were made with randomised pre-assignments, the following results were obtained:

- average run-time: 10.8 s
- standard deviation: $20,6 \mathrm{~s}$
- longest run: 2 m 52.5 s
- REGRET_MAX_MAX found solution $67 \%$ of the runs (within 10s).

The average and longest run-times are acceptable. Remember that the randomisation of pre-assignments does not correspond to real pre-assignments and thus real instances might have a lot longer run-times.

Figure 4.2 shows a generated schedule with real pre-assignments from the schedule in figure 3.1, using as needed, relaxation of constraint W8. Note when comparing with figure 3.1 that Dr. K seems to be on vacation when he/she was in fact part time available to satisfy some of the constraints adaptively, but this is not seen in the schedule. In the generated schedule Dr. K is treated as available at all times. The pre-assignments are the vacations and most of the miscellaneous tasks, some
of the miscellaneous tasks are not pre-assigned as they can be post-refined instead. This schedule is found in 13.2 s using the combined variable order.

\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|}
\hline \& \multicolumn{4}{|c|}{Week 1} \& \multicolumn{5}{|c|}{Week 2} \& \multicolumn{4}{|c|}{Week 3} \& \multicolumn{4}{|c|}{Week 4} \\
\hline Dr. A Cph \& \[
\begin{array}{ll}
\hline x \& x \\
x \& x
\end{array}
\] \& \[
\begin{array}{ll}
x \& x \\
x \& x
\end{array}
\] \& \[
\begin{array}{ll|}
\hline x \& x \\
x \& x
\end{array}
\] \& \& \[
\begin{aligned}
\& \mathrm{O} \\
\& \mathrm{O} \\
\& \mathrm{~A} \\
\& \mathrm{~A} \\
\& \hline
\end{aligned}
\] \& \(\begin{array}{ll}\text { O } \& \text { W } \\ \text { O } \& \text { W } \\ \text { A } \& \text { A }\end{array}\) \& \begin{tabular}{l|l} 
W \& O \\
W \& O \\
A \& \\
\hline
\end{tabular} \& \begin{tabular}{l|l}
\(B\) \& \(B\) \\
\(B\) \& \(B\) \\
\(B\) \& \\
\(B\)
\end{tabular} \& \& \& \[
\bar{x}
\] \& \[
\begin{array}{lll}
\hline x \& x \& x \\
x \& x \& x
\end{array}
\] \& \[
\begin{array}{ll}
\hline x \& x \\
x \& x \\
x \& x \\
\hline
\end{array}
\] \& \& \[
\begin{aligned}
\& \mathrm{O} \\
\& \mathrm{O} \\
\& \mathrm{~A} \\
\& \hline
\end{aligned}
\] \& \[
\begin{array}{lll}
\hline \mathrm{O} \& \mathrm{O} \& \mathrm{O} \\
\mathrm{O} \& \mathrm{O} \& \mathrm{O} \\
\mathrm{~A} \& \mathrm{~A} \& \mathrm{O} \\
\hline
\end{array}
\] \& \[
\begin{array}{ll}
\mathrm{x} \& \mathrm{x} \\
\mathrm{x} \& \mathrm{x} \\
\mathrm{x} \& \mathrm{x} \\
\hline
\end{array}
\] \\
\hline \& \[
\begin{array}{ll}
\hline x \& x \\
x \& x \\
x \& x
\end{array}
\] \& \[
\begin{array}{ll}
\hline x \& x \\
x \& x \\
x \& x \\
\hline
\end{array}
\] \& \[
\begin{array}{ll}
\hline x \& x \\
x \& x \\
x \& x
\end{array}
\] \& \& \[
x
\] \& \(\begin{array}{ll}x \& \mathrm{x} \\ \mathrm{x} \& \mathrm{x} \\ \mathrm{x} \& \mathrm{x}\end{array}\) \& \begin{tabular}{l} 
x \\
\hline x \\
x \\
x \\
x
\end{tabular} \& \[
\begin{array}{ll}
x \& x \\
x \& x \\
x \& x
\end{array}
\] \& \& \& \[
\begin{aligned}
\& \mathrm{O} \\
\& \mathrm{O} \\
\& \mathrm{~A}
\end{aligned}
\] \& \[
\begin{array}{lll}
\hline \mathrm{O} \& \mathrm{O} \& \mathrm{O} \\
\mathrm{O} \& \mathrm{O} \& \mathrm{O} \\
\mathrm{~A} \& \mathrm{~A} \& \mathrm{O} \\
\hline
\end{array}
\] \& \[
\begin{array}{|l|l}
\hline B \& B \\
B \& B \\
B \& B \\
\hline
\end{array}
\] \& \& \[
\begin{aligned}
\& \mathrm{O} \\
\& \mathrm{O} \\
\& \mathrm{~A}
\end{aligned}
\] \&  \& \[
x
\] \\
\hline \& \[
\begin{array}{ll}
\hline A \& O \\
A \& O \\
O \& A \\
\hline
\end{array}
\] \&  \& \[
\begin{array}{l|l}
B \& B \\
B \& B \\
B \& B \\
\hline
\end{array}
\] \& A \& \[
\begin{array}{ll}
\hline \mathrm{O} \& \mathrm{O} \\
\mathrm{O} \& \mathrm{O} \\
\mathrm{~A} \& \mathrm{~A}
\end{array}
\] \& \[
\begin{array}{l|l}
\hline \mathrm{O} \& \mathrm{O} \\
\mathrm{O} \& \mathrm{O} \\
\mathrm{~A} \& \mathrm{~A} \\
\hline
\end{array}
\] \& O \& \[
\begin{array}{|cc|}
\hline x \& x \\
x \& x \\
x \& x \\
\hline
\end{array}
\] \& \& \& \[
\begin{aligned}
\& x \\
\& x \\
\& x
\end{aligned}
\] \& \[
x \quad x \quad x
\] \& \[
\begin{array}{ll}
\hline x \& x \\
x \& x \\
x \& x \\
\hline
\end{array}
\] \& \& \[
\bar{x}
\] \& \[
\begin{array}{lll}
x \& x \& x \\
x \& x \& x \\
x \& x \& x
\end{array}
\] \& \[
\begin{array}{ll}
x \& x \\
x \& x \\
x \& x \\
\hline
\end{array}
\] \\
\hline \& \[
\begin{array}{ll}
\hline W \& A \\
W \& A \\
A \& O
\end{array}
\] \& W A W A A O \& \[
\begin{array}{ll}
x \& x \\
x \& x
\end{array}
\] \& W \&  \& \begin{tabular}{l|l|l} 
O \& W \\
O \& W \\
A \& \(A\)
\end{tabular} \& \begin{tabular}{l} 
W 0 \\
W O \\
A O \\
\hline
\end{tabular} \& \[
\begin{array}{ll|}
\hline x \& x \\
x \& x \\
x \& x \\
\hline
\end{array}
\] \& \& \& \[
\begin{aligned}
\& \mathrm{O} \\
\& \mathrm{O} \\
\& \mathrm{~A} \\
\& \hline
\end{aligned}
\] \& \begin{tabular}{|l|l|l|l}
\hline 0 \& \(O\) \& \(O\) \\
\(O\) \& \(O\) \& \(O\) \\
\(A\) \& \(A\) \& \(O\) \\
\hline \& \\
\hline
\end{tabular} \& \[
\begin{array}{lll}
\hline x \& x \\
x \& x \\
x \& x \\
\hline
\end{array}
\] \& \& \[
\begin{array}{ll}
x \& x \\
x \& x \\
x \& x \\
\hline
\end{array}
\] \&  \& \\
\hline \& \[
\begin{array}{cc}
W \& A \\
W \& A \\
A \& O
\end{array}
\] \&  \& \[
\begin{array}{ll}
x \& x \\
x \& x \\
x \& x \\
\hline
\end{array}
\] \& A \& \[
\begin{array}{ll}
\mathrm{A} \& A \\
\mathrm{~A} \& A \\
\mathrm{O} \&
\end{array}
\] \& \(\begin{array}{ll}\text { A } \& A \\ A \& A \\ \text { O } \& \text { O }\end{array}\) \& A 0
A

0 \& $$
\begin{array}{lll}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$ \& \& \& \[

$$
\begin{aligned}
& \mathrm{A} \\
& \mathrm{~A} \\
& \mathrm{O} \\
& \hline
\end{aligned}
$$

\] \& \[

$$
\begin{array}{lll}
O & W & O \\
O & W & O \\
A & A & O \\
\hline
\end{array}
$$

\] \& $\begin{array}{ll}x & x \\ x & x \\ x & x\end{array}$ \& \& \[

$$
\begin{aligned}
& \mathrm{O} \\
& \mathrm{O} \\
& \mathrm{~A} \\
& \hline
\end{aligned}
$$

\] \& | 0 | $O$ | $O$ |
| :--- | :--- | :--- |
| 0 | 0 | $O$ |
| $A$ | $A$ | $O$ | \& | $B$ | $B$ |
| :--- | :--- | :--- |
| $B$ | $B$ |
| $B$ | $B$ | <br>

\hline \& $$
\begin{array}{ll}
\hline \mathrm{A} & \mathrm{~A} \\
\mathrm{~A} & \mathrm{~A} \\
\mathrm{O} & \mathrm{O} \\
\hline
\end{array}
$$ \& \[

$$
\begin{array}{ll}
\hline O & O \\
O & O \\
A & \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{array}{ll}
\hline x & x \\
x & x
\end{array}
$$

\] \& O \&  \& \[

$$
\begin{array}{ll}
0 & 0 \\
0 & 0
\end{array}
$$

\] \& A \& \[

$$
\begin{array}{lll}
\hline x & x \\
x & x \\
x & x
\end{array}
$$

\] \& W \& \& \[

\overline{0}

\] \& \[

$$
\begin{array}{|l|ll}
\hline O & W & A \\
O & W & A \\
A & A & \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{array}{ll|}
\hline x & x \\
x & x
\end{array}
$$

\] \& \& \[

$$
\begin{array}{ll}
\mathrm{x} & \mathrm{x} \\
\mathrm{x} & \mathrm{x} \\
\mathrm{x} & \mathrm{x} \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{array}{lll}
x & x & x \\
x & x & x \\
x & x & x \\
\hline
\end{array}
$$
\] \& <br>

\hline \& | 0 | $O$ |
| :--- | :--- |
| $O$ | $O$ |
| A | A |
|  |  | \& \[

$$
\begin{array}{ll}
0 & 0 \\
0 & 0 \\
0 & \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{array}{|ll|}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$
\] \& \&  \& $\begin{array}{ll}x & X \\ x & x \\ x & x\end{array}$ \& x

x
x
x
x

x \&  \& \& \& \& $$
\begin{array}{lll}
\mathrm{x} & \mathrm{x} & \mathrm{x} \\
\mathrm{x} & \mathrm{x} & \mathrm{x} \\
\mathrm{x} & \mathrm{x} & \mathrm{x}
\end{array}
$$ \& \[

$$
\begin{array}{ll}
x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \& \[

$$
\begin{aligned}
& \mathrm{O} \\
& \mathrm{O} \\
& \mathrm{~A}
\end{aligned}
$$

\] \& \[

$$
\begin{array}{|c|cc}
\hline O & W & O \\
O & W & O \\
A & A & O \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{array}{ll}
\hline x & x \\
x & x \\
x & x
\end{array}
$$
\] <br>

\hline \& $$
\begin{array}{ll}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$ \& \[

$$
\begin{array}{ll}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{array}{ll}
\hline x & x \\
x & x \\
x & x
\end{array}
$$
\] \& \&  \& $\begin{array}{ll}x & x \\ x & x \\ x & x\end{array}$ \& X

X
x
x

x \& $$
\begin{array}{|cc|}
\hline x & x \\
x & x \\
x & x
\end{array}
$$ \& \& \& \& \[

$$
\begin{array}{ll}
\hline x & x \\
x & x \\
x & x
\end{array}
$$

\] \& \[

$$
\begin{array}{ll}
\hline x & x \\
x & x \\
x & x
\end{array}
$$
\] \& \& A

$A$
O

O \& $$
\begin{array}{|lll}
\hline A & W & A \\
A & W & A \\
O & O & \\
\hline
\end{array}
$$ \& \[

$$
\begin{array}{ll}
\hline x & x \\
x & x
\end{array}
$$
\] <br>

\hline \& $$
\begin{array}{l|l}
\hline \mathrm{A} & \mathrm{~A} \\
\mathrm{~A} & \mathrm{~A} \\
\mathrm{O} & \mathrm{O} \\
\hline
\end{array}
$$ \& \[

$$
\begin{array}{ll}
\hline \mathrm{A} & \mathrm{~A} \\
\mathrm{~A} & \mathrm{~A} \\
\mathrm{O} & \mathrm{O} \\
\hline
\end{array}
$$

\] \& \[

x \times
\] \& \&  \& $\begin{array}{ll}x & x \\ x & x \\ x & x\end{array}$ \& x

x
x
x

x \& $$
\begin{array}{ll}
\hline x & x \\
x & x \\
x & x
\end{array}
$$ \& \& \& \& \[

$$
\begin{array}{lll}
x & x & x \\
x & x & x \\
x & x & x
\end{array}
$$

\] \& \[

$$
\begin{array}{ll|}
\hline x & x \\
x & x \\
x & x
\end{array}
$$

\] \& \& \& \[

\bar{x}
\] \& x <br>

\hline \& x \&  \& \[
$$
\begin{array}{ll}
x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& | A |
| :--- | :--- |
| A |
| O | \&  \& \[

$$
\begin{array}{ll}
\hline A & A \\
A & A \\
O & O \\
\hline
\end{array}
$$

\] \& \& \[

$$
\begin{array}{ll}
x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \& \& A \& \[

$$
\begin{array}{|lll|}
\hline 0 & O & A \\
0 & O & A \\
O & O & \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{array}{ll}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \& \[

$$
\begin{array}{l|l}
\hline \mathrm{O} & \mathrm{O} \\
\mathrm{O} & \mathrm{O} \\
\mathrm{O} & \mathrm{O} \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{array}{lll}
\hline \mathrm{O} & \mathrm{O} & \mathrm{~A} \\
\mathrm{O} & \mathrm{O} & \mathrm{~A} \\
\mathrm{O} & \mathrm{O} & \\
\hline
\end{array}
$$
\] \& x x <br>

\hline \& $$
\begin{array}{ll}
x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$ \& \[

\mathrm{x} \quad \mathrm{x}

\] \& \[

$$
\begin{array}{|cc|}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$
\] \& \&  \& $\begin{array}{ll}\mathrm{X} & \mathrm{X} \\ \mathrm{x} & \mathrm{x} \\ \mathrm{x} & \mathrm{x}\end{array}$ \& X

x
x
x

x \& $$
\begin{array}{ll}
x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$ \& \& \&  \& \[

$$
\begin{array}{ll}
\mathrm{A} & \mathrm{~A} \\
\mathrm{~A} & \mathrm{~A} \\
\mathrm{O} & \mathrm{O}
\end{array}
$$

\] \& \[

$$
\begin{array}{lll}
\hline x & x \\
x & x \\
x & x
\end{array}
$$

\] \& \& \[

x

\] \& \[

\mathrm{x}

\] \& \[

x
\] <br>

\hline \& $$
\begin{array}{ll}
\hline P & x \\
P & x \\
P & x \\
\hline
\end{array}
$$ \&  \& \[

$$
\begin{array}{ll}
\hline P & x \\
P & x \\
P & x \\
\hline
\end{array}
$$

\] \& \& \[

P \quad x

\] \& \[

0 \quad 0

\] \& A A \& \[

$$
\begin{array}{ll}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \& \& \[

$$
\begin{array}{ll}
\hline x & f \\
x & 1 \\
x & 0 \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{array}{lll}
\hline A & A & A \\
A & A & A \\
O & O &
\end{array}
$$

\] \& \[

$$
\begin{array}{ll}
\mathrm{x} & \mathrm{x} \\
\mathrm{x} & \mathrm{x} \\
\mathrm{x} & \mathrm{x}
\end{array}
$$

\] \& \& \[

$$
\begin{aligned}
& \mathrm{x} \\
& \mathrm{x} \\
& \mathrm{x}
\end{aligned}
$$

\] \& \[

$$
\begin{array}{lll}
\hline A & A & A \\
A & A & A \\
O & O & \\
\hline
\end{array}
$$
\] \& <br>

\hline Dr. M Uph \& $$
\begin{array}{ll}
x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$ \&  \& \[

$$
\begin{array}{|cc|}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \&  \& \[

$$
\begin{aligned}
& \hline \mathrm{x} \\
& \mathrm{x} \\
& \mathrm{x} \\
& \hline
\end{aligned}
$$

\] \& \& \[

$$
\begin{array}{|ll}
\hline P & x \\
P & x \\
P & x \\
\hline
\end{array}
$$

\] \& \& \& \& \[

\mathrm{x}

\] \& \[

$$
\begin{array}{ll}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \& \[

$$
\begin{aligned}
& \hline P \\
& P \\
& P
\end{aligned}
$$

\] \& \[

$$
\begin{array}{ll}
\hline \text { A } & \text { A } \\
\text { A } & \text { A } \\
\text { O } & \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{aligned}
& x \\
& x
\end{aligned}
$$
\] <br>

\hline \& $$
\begin{array}{ll}
\mathrm{A} & \mathrm{P} \\
\mathrm{~A} & \mathrm{P} \\
\mathrm{O} & \mathrm{P}
\end{array}
$$ \&  \& \[

$$
\begin{array}{|cc|}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \& \[

$$
\begin{array}{lll}
\hline A & A & F \\
A & A & F \\
O & O & F \\
\hline
\end{array}
$$

\] \& P \& \& \[

$$
\begin{array}{lll}
\hline x & x \\
x & x \\
x & x
\end{array}
$$

\] \& \& O \& \[

$$
\begin{aligned}
& \hline P \quad, \\
& P \text {, } \\
& P \text {, }
\end{aligned}
$$

\] \& \[

$$
\begin{array}{ccc}
\hline \mathrm{x} & \mathrm{~A} & \mathrm{~A} \\
\mathrm{x} & \mathrm{~A} & \mathrm{~A} \\
\mathrm{x} & \mathrm{O} & \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{array}{ll}
\hline P & x \\
P & x \\
P & x \\
\hline
\end{array}
$$

\] \& \& \[

0

\] \& \[

$$
\begin{array}{ll}
P & x \\
P & x \\
P & x \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{aligned}
& \mathrm{x} \\
& \mathrm{x} \\
& \mathrm{x}
\end{aligned}
$$
\] <br>

\hline Dr. 0 Uph \& $$
\begin{array}{ll}
0 & 0 \\
0 & 0 \\
0 & 0
\end{array}
$$ \& \[

$$
\begin{array}{ll}
\hline P & x \\
P & x \\
P & x \\
\hline
\end{array}
$$

\] \& \[

\bar{x}

\] \& \&  \& \& \& \[

$$
\begin{array}{ll}
x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \& \& \[

\mathrm{O}

\] \& \[

$$
\begin{array}{ll}
\hline P & x \\
P & x \\
P & x \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{array}{ll}
x & x \\
x & x
\end{array}
$$

\] \& \& \[

$$
\begin{array}{ll}
\hline W & W \\
W & W \\
O & 0
\end{array}
$$
\] \& W W W

$W$

W \& $$
\begin{aligned}
& P x \\
& P x \\
& P x
\end{aligned}
$$ <br>

\hline Dr \& $$
\begin{array}{ll}
0 & 0 \\
0 & 0 \\
0 & 0
\end{array}
$$ \&  \& \[

$$
\begin{array}{|cc|}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \&  \& \[

$$
\begin{aligned}
& \hline W W \\
& W \\
& W \\
& O
\end{aligned}
$$

\] \& W \& \[

$$
\begin{array}{ll}
\hline x & x \\
x & x \\
x & x
\end{array}
$$

\] \& \& O \& \[

0

\] \& \[

$$
\begin{array}{lll}
\hline \text { W } & \text { W } & \text { W } \\
\text { W } & \text { W } & \text { W } \\
0 & &
\end{array}
$$

\] \& \[

$$
\begin{array}{ll|}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \& \[

$$
\begin{array}{ll}
\hline A & A \\
\text { A } & A \\
O & O
\end{array}
$$

\] \& \[

$$
\begin{array}{lll}
\hline A & A & P \\
A & A & P \\
O & & P \\
\hline
\end{array}
$$
\] \& <br>

\hline \& $$
\begin{array}{ll}
\hline 0 & 0 \\
0 & 0 \\
& 0 \\
\hline
\end{array}
$$ \& \[

$$
\begin{array}{ll}
\hline O & P \\
O & P \\
& P
\end{array}
$$

\] \& \[

$$
\begin{array}{ll|}
\hline x & P \\
x & P \\
x & P \\
\hline
\end{array}
$$

\] \& \&  \& \[

0

\] \& \& | $x$ | $x$ |
| :--- | :--- |
| $x$ | $x$ |
| $x$ |  | \& \& \& \[

0

\] \& | W W | W |  |
| :--- | :--- | :--- |
| W | $W$ | $W$ |
| 0 |  |  |
|  |  |  | \& \[

$$
\begin{array}{ll}
\mathrm{x} & \mathrm{x} \\
\mathrm{x} & \mathrm{x} \\
\mathrm{x} & \mathrm{x} \\
\hline
\end{array}
$$

\] \& \& \[

\mathrm{x} \quad \mathrm{x}
\] \&  \& <br>

\hline Dr. R Uph \& $$
\begin{array}{ll}
\hline W & W \\
W & W
\end{array}
$$ \& W W W ${ }_{\text {W }} \mathrm{W}$ \&  \& \&  \& \[

$$
\begin{aligned}
& \mathrm{O} \\
& \mathrm{O} \\
& \mathrm{O}
\end{aligned}
$$

\] \& \& \[

$$
\begin{array}{ll|}
\hline x & P \\
x & P \\
x & P \\
\hline
\end{array}
$$

\] \& \& \& \[

\mathrm{O}

\] \& \[

$$
\begin{array}{lll}
\hline O & P & x \\
O & P & x \\
O & P & x
\end{array}
$$

\] \& \[

$$
\begin{array}{ll}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \& \[

$$
\begin{array}{ll}
\hline W & W \\
W & W \\
0 \\
\hline
\end{array}
$$

\] \& | W W W |  |
| :--- | :--- |
| W W W |  |
| 0 |  | \& <br>

\hline Dr. S Uph \& $$
\begin{array}{ll}
\hline W & W \\
W & W
\end{array}
$$ \& W W \& \[

$$
\begin{array}{|cc|}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \&  \& W \& W \& \[

$$
\begin{array}{|cc|}
\hline x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$

\] \& \&  \& \[

\mathrm{O}

\] \& \[

$$
\begin{array}{ll}
\hline \mathrm{O} & \mathrm{P} \\
\mathrm{O} & \mathrm{P} \\
& \mathrm{P} \\
\hline
\end{array}
$$

\] \& \[

$$
\begin{array}{ll|}
\hline x & P \\
x & P \\
x & P \\
\hline
\end{array}
$$

\] \& \& \[

$$
\begin{array}{ll}
\hline x & A \\
x & A \\
x & O \\
\hline
\end{array}
$$

\] \&  \& \[

$$
\begin{array}{ll}
x & x \\
x & x \\
x & x \\
\hline
\end{array}
$$
\] <br>

\hline
\end{tabular}

Figure 4.1: Generated schedule with randomly pre-assigned vacations using MAX_MAX.

|  | Week 1 |  |  |  |  |  |  | Week 2 |  |  |  |  |  |  | Week 3 |  |  |  |  |  |  | Week 4 |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | X | $x$ x | x | X | x | $x$ | x | X | x | X | X | X | x | x | X | x | x | X | x |  |  |  | $x$ x | x | X | $x$ | $x$ x |
| Dr. A | X | $x$ x | x | X | X | x | x | X | X | x | X | X | X | x | X | x | x | X | X | $x$ | $x$ x |  | x x | $x$ | X | X | $\mathrm{x} \times$ |
| Cph | x | x x | x | X | x | x | x | x | x | X | X | $x$ | x | x | X | x | x | X | X | $x$ | x |  | x | X | X | X | x x |
|  | W | W O | 0 O | W | x | X | X | x | x | x | x | x | X | x | X | x | x | 0 | A | x | x |  | 00 | 0 | 0 | O | $x$ x |
| Dr. B | W | W O | 0 O | W | x | x | x | x | x | x | X | X | x | x | x | $x$ | x | 0 | - |  | x |  | 00 | 0 | 0 | 0 | X x |
| Cph | A | A A | A | A | X | x | x | X | X | X | X | X | $x$ | $x$ | X | x | x | - | 0 | x | x |  | - - | A | A | 0 | $x$ x |
|  |  | W A | A A | W | A |  |  | W | A | 0 | W | X | X |  | 0 | O | 0 | O | x |  | X X |  | O | O | O | X | X X |
| Dr. C | W | W A | A A | W | A | $x$ |  | W | A | 0 | W | $x$ | $x$ | x | 0 | 0 | 0 | 0 | $x$ | $x$ | $x$ x |  | 0 | 0 | 0 | x | $x$ x |
| Cph | - | 0 | 0 | A | 0 | x |  |  | - | A | A | x | $x$ | x | - | A | A | A | x | X | X x |  | x | A |  | x | x x |
|  | A | A A | A A | A | $x$ | X | $x$ | W | A | A | W | A | $x$ | x | W | O | O | W | 0 | x | $x$ |  | 0 | 0 | 0 | O | $x \times$ |
|  | A | A A | A A | A | $x$ | x |  | W | A | - | W | A | $x$ |  | W | O | 0 | W | O |  | $x \times$ |  | 00 | 0 | 0 | O | $x$ |
| Cph | 0 | 0 | 0 | 0 | x | X | x | A | 0 | A | - | 0 | x | x | A | A | - | - | 0 | X | x |  | A | A |  | 0 | x x |
|  | A | A 0 | 0 |  |  | X | $x$ | 0 | 0 |  |  |  | x | X | - | O | - | 0 | x | $x$ | $x$ |  | W O | O | W | O | x x |
| Dr. E | A | A 0 | 0 |  |  | x | $x$ | 0 | 0 |  |  |  | x | x |  | 0 | - | O | $x$ | x | $\times$ x |  | W | O | W | 0 | x x |
| Cph | 0 | 0 A | A |  |  | X | x | A | - | - | - |  | x | x | - | A | A | A | x | X | x |  | A A | A | A | 0 | x x |
|  | X | x X | x | x | X | x | x | x | x | X | X | X | x | x | X | X | X | X | X | X | $x$ x |  | x x | X | X | X | X x |
| Dr. F | X | x X | $x$ | X | $x$ | $x$ | x | x | x | x | X | $x$ | $x$ | $x$ | x | x | x |  | x | $x$ | $x$ x |  | $x$ x |  | x | x | x x |
| Cph | X | x | x | X | x | X | x | X | X | X | x | X | X | $x$ | x | X | X | x | X | X | x x |  | X X | X | X | x | $x \mathrm{x}$ |
|  | A | A A | A A | A | 0 | x | $x$ | A |  |  | 0 | 0 | x | $x$ | W | 0 | 0 | W | O |  | x |  | W | O | W | X | $x$ x |
| Dr. G | A | A A | A A | A | O | x |  | A |  | A | O | O | x | x | W | O | O | W | O | x | $x$ |  | W O | O | W | x | $x$ x |
| Cph | 0 | 0 O | 0 O |  | 0 | X | x | - | - | 0 | - |  | X | $\times$ | A | A | A | - | 0 | x |  |  | A A |  |  | X | $\mathrm{x} \times$ |
|  | A | A A | A A | A | x | x | x | X | X | x | x | X | x | $x$ | 0 | 0 | O | 0 | X | X | x |  | 00 | 0 | O | X | X x |
| Dr. H | A | A A | A A | A | $x$ | $x$ | x | X | x | x | x | X | X | x | 0 | 0 | 0 | O | X | x | $x$ |  | 0 | 0 | O | X | x x |
| Cph |  | 0 | 0 | 0 | x | x | x | x | x | X | X | x | x | $\times$ |  | O | A | A | x | x |  |  | A A |  | A | x | $\mathrm{x} \times$ |
|  | 0 | 0 | 0 | 0 | A | $x$ | $x$ | A | A | A | A | - | x | $x$ | A | A | A | A | 0 | x | $x$ |  | 0 | A | 0 | A | $x \times$ |
| Dr. I | 0 | 0 | 0 | 0 | A | $x$ | x | A | A | A | A |  | x | x | A | A | A | A | 0 | X | x x |  | 0 | A | 0 | A | $x \mathrm{x}$ |
| Cph | A | A A | A A | - | 0 | x | $x$ | O | 0 | - | - | - | X | x $\times$ | 0 | 0 | 0 | 0 |  | X | x |  | 0 A | O | A |  | $\mathrm{X} \times$ |
|  | X | x x | x | X | X | X |  | A | A | A | 0 | O | x |  |  | O | O | - | A | $x$ | x |  | 0 A | A | O | A | $\mathrm{X} \times$ |
| Dr. J | x | x X | $x$ | x | x | $x$ | x | A | A | A | 0 | 0 | X | x | 0 | 0 | 0 | 0 | A | x | x |  | 0 A | A | 0 | A | $x$ x |
| Cph | X | x | x | X | X | x | $x$ | - | 0 | 0 | - | 0 | X | x |  | 0 |  |  |  |  | $\times$ |  | O |  |  |  |  |
|  |  |  | 0 | 0 | A | $x$ |  | A | 0 | 0 | A | A | $x$ |  | A | 0 | 0 | A | A | $x$ |  |  | A A | A | A |  |  |
| Dr | O | 0 | 0 O | 0 | A | $x$ | $\times$ | A | 0 | 0 | A | A | $x$ |  | A | O | O | A | A | $x$ | $x \times$ |  | A A | A | A |  |  |
| Cph | A | A A | A A | A |  | X | $\times$ | 0 | A | A | O | O | $x$ |  | 0 | O | O | 0 |  | x |  |  | 0 | O | 0 |  | X x |
|  |  |  | W O | P | x | x | $x$ | W | O | A | P | x | x | $x$ | W | W | W | W | P | x | P |  | x W | W | W | W | x x |
| Dr. L | - | W | W O | P | X | X | $x$ | W | 0 | - | P | X | X | x | W | W | W | W | P |  | P |  | W | W | W | W | x x |
| Uph | - | 0 | 0 A | P | X | X | x | A | - | - | P | x | x | x | A | O | - | 0 | P | x | P |  | X | 0 |  |  | X x |
|  | X | X X | X X | X | X | X | x | W | W | W | W | P |  |  | X | A | A | P | X | X | x |  | W A | A | P | X | X x |
| Dr. M | X | X X | x | X | X | X | x | W | W | W | W | P | x | P | x | A | A | P | x | $x$ | $\times \times$ |  | W A |  | P | X | X x |
| Uph | X | X | X | X | X | X | x | - | - | A | A | P | X | P | X | 0 | - | P | X | X | x x |  | 00 | 0 | P | X | X x |
|  | 0 | 0 P | P | 0 |  | P | x | 0 | P | x | A | W | x | x | P | x | W | W | W | x | x |  | A | x | A | W | $x$ x |
| Dr. N | O | 0 P | P | 0 |  | P | x | 0 | P | x | A | W | x | x | P | x | W | W | W | x | x |  | - P | x | A | W | x x |
| Uph | A | A P | P | 0 | - | P | x | A | P | x | O |  | X |  | P | x |  |  |  | x | $x \times$ |  | P | x | 0 |  | $\mathrm{x} \times$ |
|  | X | X X | $x$ X | X | X | X | $\times$ | O | O | - |  |  | P |  | A | P | x | A | A | $x$ | x |  | P |  | A | A | $\mathrm{x} \times$ |
| Dr. 0 | x | $x$ x | $x$ x | x | x | $x$ | $x$ | 0 | O |  |  |  | P | x | A | P | $x$ | A | A | X | x |  | P P | A | A | A | $x \quad x$ |
| Uph | X | x | $\mathrm{x} \times$ | X | X | X | $x$ | 0 | A | - | - |  | P |  | - | P | x | - | - | $x$ | $x$ |  | P X | 0 |  |  |  |
|  | 0 | 0 | P | x | W | x | x | P | X | 0 | 0 | A | x | x | X | X | x | X | x | X | X x |  | x - | - | A | A | P |
| Dr. P | 0 |  | - P | x | W | x | x | P | x | 0 | O | A | x | x | X | x | x | x | X | x | $x$ x |  | A |  | A |  |  |
| Uph | O | 0 | 0 P | x | - | x | $x$ | P | x | - | A | - | X | $x$ | X | x | X | x | x | X | X x |  | 0 |  |  | - |  |
|  | W | W W | W W | W | P | x | P | X | W | W | W | W | x |  | W | W | P |  | W | x |  |  | x x | X | x | X |  |
| Dr. Q | W | W W | W W | W | P |  | P | x | W | W | W | W | x |  | W | W | P | x | W | x | x |  | $x$ x | x | x | x | X x |
| Uph | 0 | 00 | 0 O | O | P | x | P | x | A | - | O | - | x | x | 0 | O | P | x | - | X | x |  | $x$ x | X | x | x | X x |
|  | W | W O | O W | W | W | X | $x$ | 0 | O | P | X | A | X |  | A | A | A | - | - |  | $x$ |  | A | P | x |  |  |
| Dr. R | W | W O | O W | W | W | x |  | 0 | 0 | P | x |  | x |  | A | A | A | - | - |  | x |  |  | P | x |  | X x |
| Uph |  |  | 0 |  |  |  |  | 0 | 0 | P | x |  |  |  | 0 |  | - |  |  |  |  |  | 00 | P | X | - |  |
|  | P | P | x | X | X | X | x | X | X | X | X | X | X | X | X | x | X | x | X | x | x |  | W W | W | W | P | x P |
| Dr. S | P | P x | x | X | x | X | x | X | x | X | X | x | X | x | X | x | x | x | x | x | x |  | W W | W | W | P |  |
| Uph | P | X | x 0 | x | X | X | x | X | x | X | X | X | X | x | X | x | X | x | X | x | x |  | 0 | 0 | 0 | P | X P |

Figure 4.2: Generated schedule with real pre-assignments.

## Chapter 5

## Conclusions

This thesis discussed constraint programming for scheduling and presents a model for a clinic in Sweden. The starting point was that the clinic expressed interest in using software for their needs and from this, many aspects comes up. The model that I tried to implement in the beginning was HIBISCUS [1], which uses many global constraints and has been successfully implemented to several different contexts. However, in the Kalmar case, the rules were not as sophisticated and the main topic in the problem here seems to be a quite large instance.

Extensive tests with random pre-assignments were carried out that gave guidance to what variable order is good for this type of model. The only global constraints that were used in the non-relaxed model are: count and logic constraints. The scheduling software is able to generate schedules for the tested instance from Kalmar in 13.2 s .

### 5.1 Restrictions and Delimitations

The model is based on pre-assignments that need to be given by a doctor and the number of instances tested are small. All doctors have been divided into two different groups, in reality the constraints differ more than that and constraints on individuals would be useful. Like with the pre-assignments the definition of such constraints is to be given by a user and for this, a good graphical interface is needed.

Since there are no precise general rules in constraint programming that state what model gives the best performance, it is important to test several different ones empirically. Since both constraints, pre-assignments, variable and value order and schedule model all can be altered, the number of combinations of tests is big. Changing the value order might give better performance for some instances.

The scheduling horizon is one month, even though the manually constructed schedules are done with a 2 month horizon. It is possible to search for the schedule one month at a time, since the constraints only span over that horizon. It is preferable to use the whole horizon however, since it leaves open for more flexibility and more constraints, should they occur.

The schedule model uses a 3D-matrix which yields a large number of variables. Changing to assigning one value for each day, as described in 3.1, yields a model with: 560 variables and 19 values ( 38 with the back-up on-call included in the main schedule ), compared to 1608 variables and 6 values in the current model. Using day variables could give a lot better performance and is indeed interesting to test. It will however make the constraints less intuitive and the model will be less dynamic when it comes to changing constraints, since more constraints are implied by the model.

### 5.2 Future Directions

No optimisation techniques have been considered in this paper. However, looking at the Kalmar case, the set of constraints that were given, were not satisfiable when using the real sets of pre-assignments. A common way is to optimise a linear objective function on the number of satisfied soft constraints. These weights must somehow be specified by the clinic.

An attempt to find implied constraints in the model was made, but none was found. Tie-breaking in the variable order as well as local search are other techniques that have not been used.

## Acknowledgements

I would like to thank my supervisor, professor Pierre Flener, for all the help and support during the work of this thesis and for being an inspiring lecturer in the Constraint technology course that got me interested in the subject. I also thank Patrik Bernestrå, doctor and scheduler at the orthopaedics department in Kalmar, for providing data and constraints for this topic and for his continuous help explaining how physicians work and the difficulties of scheduling.

## References

[1] Stéphane Bourdais, Philippe Galinier, and Gilles Pesant. Hibiscus: A constraint programming application to staff scheduling in health care. In Proceedings of CP’03, pages 153-167. Springer-Verlag, 2003.
[2] P. Baptiste, C. Le Pape, and W. Nuijten. Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems. Kluwer Academic Publishers, 2001.
[3] G. Laporte. The art and science of designing rotating schedules. In Proceedings of the Journal of the Operational Research Society, volume 50, pages 1011-1017, October 1999.
[4] Nicolas Beldiceanu, Mats Carlsson, and Jean-Xavier Rampon. Global constraint catalog. Technical Report T2005-08, Swedish Institute of Computer Science, February 2008. See http://www.emn.fr/xinfo/sdemasse/gccat/index.html.
[5] Slim Abdennadher and Hans Schlenker. Interdip - an interactive constraint based nurse scheduler. In Proceedings of The First International Conference and Exhibition on The Practical Application of Constraint Technologies and Logic Programming, PACLP, 1999.
[6] Michael W. Carter and Sophie D. Lapierre. Scheduling emergency room physicians. In Proceedings of Health Care Management Science 4, pages 347-360, 2001.
[7] Gecode Team. Gecode: Generic constraint development environment, 2008. Available from http://www.gecode.org.
[8] S. J. Darmoni, A. Fajner, N. Mahe, A. Leforestier, M. Vondracek, O. Stelian, and M. Baldenweck. Horoplan: computer-assisted nurse scheduling using constraintbased programming. In Proceedings of Journal of the Society for Health Systems, pages 5:41-54, 1995.
[9] Harald Meyer Auf'm Hofe. Nurse rostering as constraint satisfaction with fuzzy constraints and inferred control strategies. In Proceedings of DIMACS workshop
on Constraint programming and large scale discrete optimization, pages 67-100, Boston, MA, USA, 2001. American Mathematical Society.
[10] M. Kelishani and G. Lindmark. Time care - schemaläggningsprogram med individuella och flexibla arbetstider: en vinn-vinn situation för arbetsgivare och anställda? Master's thesis, Luleå University of Technology, Department of Human Work Sciences, 2005.


[^0]:    Handledare: Pierre Flener
    Ämnesgranskare: Justin Pearson
    Examinator: Tomas Nyberg
    ISSN: 1401-5757, UPTEC F08 064

[^1]:    ${ }^{1}$ See [4] for a comprehensive list of global constraints and their definitions.

