Aalto University
School of Science
Degree Programme in Engineering Physics and Mathematics

Teemu Kinnunen

Cost-efficient vacation planning with variable workforce demand and manpower

Master's thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology in the Degree Programme in Engineering Physics and Mathematics.

Helsinki, September 26, 2016

Supervisor: Professor Enrico Bartolini
Instructor: D.Sc. Antti Punkka

Aalto University
School of Science
Degree Programme of Engineering Physics and Mathematics

ABSTRACT OF MASTER'S THESIS

Author:	Teemu Kinnunen		
Title:	Cost-efficient vacation planning with variable workforce demand and manpower		
Date:	September 26, 2016	**Pages:**	iv + 64
Professorship:	Systems and operations research	**Code:**	F3008
Supervisor:	Professor Enrico Bartolini		
Instructor:	D.Sc. Antti Punkka		

Vacation planning can be a complicated process as multiple law and contract based rules must be respected, while at the same time the wishes of employees must be taken into account. The problem is especially difficult in transit industry, where demand and available manpower can vary and the products of transit industry have no shelf life. Also, temporary workers cannot be recruited as long training is needed.

In this thesis, a constraint programming formulation for solving vacation planning problems is developed. Constraint programming allows modeling each vacation as a single interval variable. This makes the approach more effective than modeling the problem as MILP, which would require a large amount of additional constraints and variables to model the problem, especially the consecutiveness of vacations. The objective of vacation planning is to find a solution, which has as large as possible minimum reserve of employees after all vacations are assigned. An additional objective of minimizing maximum reserve is introduced to even out the distribution of reserve. The problem is solved to optimality with a commercial optimization solver with running times varying from a few seconds to three minutes. The results of two real world cases of a transportation company show that the model provides improvement in solution quality and the planning time needed is reduced considerably.

The issue of planning vacations has received little attention in literature. In many cases the vacations are planned by mutual agreement or a named employee assigns vacations by hand. This can result in a lot of manual labor after which the solution quality might still be poor. This thesis presents the first constraint programming based approach for planning employees' vacations. It allows the modeling of multiple constraints that are used to improve solution quality, and takes into account the preferences of the employees, the planning personnel and the company.

Keywords:	vacation planning, scheduling, constraint programming, variable demand, rail transportation, interval variable
Language:	English

Aalto-yliopisto
Perustieteiden korkeakoulu
Teknillisen fysiikan ja matematiikan tutkinto-ohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Teemu Kinnunen		
Työn nimi:	Lomien kustannustehokas suunnittelu vaihtelevan kysynnän ja vaihtelevan työvoiman määrän tilanteessa		
Päiväys:	26. syyskuuta 2016	**Sivumäärä:**	iv + 64
Professuuri:	Systeemi- ja operaatiotutkimus	**Koodi:**	F3008
Valvoja:	Professori Enrico Bartolini		
Ohjaaja:	TkT Antti Punkka		

Lomien suunnittelu voi olla hankala prosessi, koska lain ja työehtosopimuksen asettamia rajoitteita pitää kunnioittaa ja samalla työntekijöiden toiveet pitää ottaa huomioon. Ongelma on erityisen hankala kuljetusalalla, koska kysyntä ja työvoiman määrä voivat vaihdella, ja kuljetuksia ei voi laittaa varastoon. Lisäksi väliaikaisia työntekijöitä ei voida palkata vaadittavan pitkän koulutuksen vuoksi.

Tässä työssä kehitetään rajoiteohjelmointimalli (engl. constraint programming), jota käytetään lomien suunnitteluongelman ratkaisemiseen. Rajoiteohjelmointi mahdollistaa yksittäisen loman mallintamisen yhtenä intervallimuuttujana. Tämä tekee lähestymistavasta paljon tehokkaamman kuin ongelman mallintaminen MILP-tehtävänä, mikä vaatii monia lisärajoitteita ja –muuttujia, erityisesti lomien yhdenjaksoisuuden mallintamiseksi. Lomien suunnittelussa on tavoitteena tuottaa ratkaisu, jossa on mahdollisimman suuri minimityöntekijäreservi lomien kiinnittämisen jälkeen. Lisätavoitteena otetaan käyttöön suurimman reservin minimointi, mikä tasoittaa reservin ajallista jakautumista. Ongelma ratkaistaan optimiin kaupallisella optimointiohjelmistolla ja ratkaisuajat vaihtelevat muutamista sekunneista kolmeen minuuttiin. Kaksi oikeaan dataan perustuvaa esimerkkitapausta näyttävät, että kehitetty malli parantaa tulosten laatua ja vähentää huomattavasti lomien suunnitteluun tarvittavia työtunteja.

Lomien suunnittelu on saanut vain vähän huomiota kirjallisuudessa. Monissa tapauksissa lomat suunnitellaan yhteisellä sopimisella tai yksi työntekijä suunnittelee käsin kaikkien lomien ajankohdat. Tämä voi vaatia paljon manuaalista työtä ja silti tulosten laatu voi olla huono. Tässä tutkielmassa esitetään ensimmäinen rajoiteohjelmointiin perustuva lähestymistapa työntekijöiden lomien suunnitteluun, mikä mahdollistaa useiden ratkaisujen laatua parantavien rajoitteiden mallintamisen, ottaen huomioon työntekijöiden, henkilöstösuunnittelijoiden ja työnantajan preferenssit.

Asiasanat:	lomien suunnittelu, aikataulutus, rajoiteohjelmointi, vaihteleva kysyntä, rautatieliikenne, intervallimuuttuja
Kieli:	Englanti

Contents

1. Introduction

1.1. Background and motivation

The Finnish railway operator VR Group (VR, for short) has about 9500 employees and a turnover of 1421.1 million euros. Today, VR operates around 1500 trains daily, which consists of 300 long-distance trains, 850 commuter trains and 350 freight trains. Most of VR's operations are based in Finland, but the company also has some operations in other countries, especially in Russia and Sweden. The company has three core business sectors: passenger services, logistic services and infrastructure engineering.

The operating environment in Finland is different from most of the European countries, as the Baltic Sea isolates Finland from the rest of the European rail network and the track gauge differs from the EU standard. On the other hand, the Finnish railways are well connected to Russia, which has similar track gauge. The winter conditions are challenging with snow causing various difficulties for rail network and rolling stock. In addition, the temperatures can drop to -40 °C. As over 90 % of the network is single tracked* and the traffic is dense, the capacity and flexibility are limited which makes the railway operations vulnerable to delays.

In the current increasingly competitive environment, companies need to operate as cost-efficiently as possible. Personnel costs are usually one of the largest expenses, and can be influenced by personnel resource planning. Resource planning at VR includes planning duties for train drivers, allocating duties to rosters and planning vacations at an annual level. Each of these areas provides opportunities to cost savings. Efficient duty planning i.e. allocation of driving and shunting tasks to duties reduces total working time. With efficient roster planning duties are allocated to drivers so that the drivers' working time limit is utilized in its entirety without the need of overtime. Finally, with cost-efficient vacation planning changes in personnel demand can be taken into account, which further reduces the overtime costs.

*www.vrgroup.fi/en, visited September 16, 2016

Personnel scheduling problems are widely studied and relevant in most of the companies in various industries. The problems are especially difficult in transit industry, as there are multiple characteristics that complicate personnel scheduling, e.g. irregular working hours, variable demand and variable manpower available. Also, the demand is immediate; services of transit companies cannot be stored in shelves such as manufactured goods (Koutsopoulos and Wilson 1987). Personnel scheduling consists of multiple sub-problems, one of which is vacation planning.

1.2. Objective and scope

The focus of this thesis is vacation planning of train drivers. The goal of vacation planning is to assign vacations of each worker so that there is as much as possible reserve on each day, and all law and contract based rules are fulfilled. The reserve is important, as it allows room for sick leaves and other absences. The reserve also makes solutions more robust for example in case of traffic forecasts; if the actual demand is slightly higher than the forecast, reserve workers reduce the need of costly overtime work.

For drivers, the most important issue in vacation planning is circulation of vacations. For example if a driver had the previous year's summer vacation at the beginning of June, the next summer vacation should start in mid to late July, moving forward at least 6 weeks. Some of the other constraints for planning include the length of vacations, the order of vacations between drivers, maximum number of drivers on vacation and dates, when a vacation cannot start.

The objective of this thesis is to develop an optimization model for vacation planning of train drivers. Previously vacation planning at VR has been made manually using Excel tables, so creating an automated model would greatly reduce the time needed for planning. In addition to reducing planning time, the model should produce more cost-efficient solutions than the previous method. The model will be tested using real world data in order to ensure that it can be implemented to actual vacation planning.

The scope of the thesis is limited to planning vacations for train drivers. Vacation planning for other employees such as conductors or office workers is not considered, although the developed model could easily be adapted to different employees and constraints. As vacation planning is made months before the actual vacation season starts, all drivers will not know, when they need days off. Drivers can request days off after the vacation planning, but granting or denying those request is out of the scope of this thesis. Planning personnel decide if the requests can be fulfilled when they make the rosters for the drivers, which is approximately one month before the requested day offs.

1.3. Structure of the thesis

The rest of the thesis is structured as follows. The relevant literature including publications about vacation planning, days-off optimization and the use of constraint programming in scheduling is reviewed in chapter 2. The vacation planning problem is presented in detail in chapter 3, including a constraint programming formulation of the problem. Chapter 4 introduces example cases of using the developed vacation planning model. Experiences of the use of the model in planning actual vacations are detailed in chapter 5. Finally, chapter 6 concludes the thesis and discusses issues that can be addressed in future research.

2. Theoretical background

2.1. Literature reviews

There are only a few papers that cover vacation scheduling, but there are multiple literature reviews about scheduling and rostering in general. A roster is a plan that includes all the duties that an employee must perform in a planning period which has length of usually 2-4 weeks. The literature reviews cover some publications about vacation scheduling and closely related days-off optimization which has received considerably more attention in the literature. Ernst et al. (2004b) provide a broad overview of personnel scheduling and rostering problems. The authors highlight multiple benefits of personnel scheduling optimization. Optimization allows an organization to meet customer demands cost-effectively while simultaneously respecting multiple other criteria such as shift equity, staff preferences, and flexible workplace agreements. Developing mathematical models and algorithms for a rostering tool involves three phases: demand modeling, choosing solution technique(s), and the specification of a reporting tool. Every industry has unique characteristics which make beneficial to develop tailored mathematical models and algorithms. The authors also present a classification of personnel scheduling and rostering problems, which serves as a general framework for classifying the related publications. Ernst et al. (2004a) use the framework for presenting an extensive bibliography of personnel scheduling and rostering studies including over 700 references.

A more recent literature review is provided by Van Den Bergh et al. (2013). The focus of the review is on articles that are published after 2004. The authors bring up the issue that most of the papers focus purely on creating feasible schedules for workers, and the personnel scheduling problem is rarely integrated with other scheduling problems such as machine- or operating room scheduling. Also, the multiple problems related to personnel scheduling (e.g. forecasting workload, hiring/firing, planning trainings and taking into account employees' preferences for holidays) are rarely integrated. The authors note that *constraint programming* methods are very appropriate for personnel scheduling, as these problems are often highly constrained.

A comprehensive description of the different phases of personnel scheduling in a hotel environment is presented in the four papers of Thompson (1998a, 1998b, 1998c, and 1998d). The process starts with forecasting work demand, continues with creating duties and schedules for workers, and ends with real-time control of the schedules.

2.2. Vacation planning

Vacation planning can be viewed as a sub-problem of a more general personnel scheduling problem. When a crew scheduling process is described, however, vacation planning is often ignored or left with little attention. This is possibly because in many industries vacation planning is a fairly simple task as temporary workers are available, workload is approximately constant in time, and new employees can be recruited to replace retired workers (Gärtner et al., 1998). However, some industries have characteristics that make vacation planning more difficult. For example, there may be unavailability of temporary personnel, long trainings that precede recruitments, fluctuating demand between and within seasons, and varying maximum (non-overtime) working hours between planning periods.

Koutsopoulos and Wilson (1987) highlight the importance of vacation scheduling. They focus on workforce planning in the transit industry which has many of the characteristics that make vacation scheduling difficult. The services of transit industry have no shelf-life and demand is immediate. Service reliability is very important while the amount of work and the available manpower on each day is uncertain. Moreover, lengthy and specialized training is needed before a person can work as a driver. The authors present an integrated framework for operator workforce planning which includes planning on three different levels: strategic, tactical and operational.

Their strategic model uses a planning horizon which is longer than one year and plans are made at weekly or monthly level. It models decisions about workforce size, hiring levels and vacation allocation. The authors propose a model which incorporates all the strategic level decisions. The described problem is similar to an inventory control problem where employees are interpreted as the inventory. The objective function is to minimize the annual operating costs including salaries, benefits and overtime. As many optimal solutions are possible, secondary objectives which include determining the ideal vacation levels and hiring levels i.e. the number of employees on vacation and to be hired, are also introduced. These levels are predetermined, and the authors do not discuss how they are determined. A tactical model uses the length of the timetable as the planning horizon and produces plans at the daily level. The model allocates days off and leaves an optimal number of employees on duty. An operational model is used daily to assign duties for reserve workers.

Chong and Strevell (1985) present an auction based method for distributing vacations of military flight crews. The goal of the method is to fulfill the wishes of the crews as much as possible. Each crew gets a number of points which can be used to place offers for desired vacation times. The constraint that has to be satisfied when giving the vacations for crews is that there is a certain demand for workforce in each time period which determines the minimum number of crews that must be at work in that period.

First the crews mark their desired vacation times in a calendar which is visible to everyone. As crews see what vacation times the others desire; they can move their desired vacation times to periods with fewer requests. In the second phase, the times of vacations are fixed, and crews can place offers for the vacation times they have chosen (each crew has 20 points for placing offers). Each vacation time is granted to the crew that has made the greatest offer. Other crews who have bid for the same time get their points back and can use them to bid for vacation times that are still free. This method gives each crew an equal chance to get a vacation time that they want, and distributes the requests for vacation times more evenly through the year. After implementing the new method for distributing the vacations, the crews' satisfaction was increased considerably.

Dewess (2010) focuses on vacation planning in a German public transport company. He considers the problem of planning the annual holidays while taking into account legal constraints, company issues and the preferences of drivers. German local public transport companies have up to 10 000 drivers and each driver has around 30 days of annual holiday. Vacation planning is a time-consuming task, if computer aided systems are not used. Manually created solutions are suboptimal and can violate some constraints. The author proposes a new model for vacation planning. First, drivers apply for a vacation in their preferred dates. Each application includes a minimum duration, a maximum duration, an earliest beginning date and a latest end date of the vacation. A point system is suggested in order to make vacation requests for popular days more expensive than those for other days. Often, the drivers' preferences for holiday dates are very rigid and conflicts can occur frequently even if conflict avoiding measures are adopted.

Each vacation request is assigned a score, which measures company issues, driver issues and social acceptability. The score is calculated for each application that is approved and a penalty is given for each application that is denied in the schedule. Binary decision variables are used to indicate, if an application is approved or not. The objective of vacation planning is to maximize the total benefit, where penalties of denied vacations are subtracted from scores of accepted vacations. The problem is solved with a two-stage heuristic approach. First a conflict resolution heuristic is utilized to get a feasible solution. After that iterative repair heuristic is used to improve the solution (allow more applied holidays to be granted). The algorithm produces good solutions with real world data; both decision makers and workers preferred generated solution in comparison to previous manually created one.

Gärtner et al. (1998) discuss how vacations can be taken into account, when designing rosters. On average 15 % of the Austrian employees are on sick leave, vacation, or other absence. The timing of the absences is only partially known. The authors evaluate multiple techniques for dealing with vacations including company vacations (i.e. all or most of the employees have vacation at the same time), postponing less important tasks, using temporary workers, increasing the amount of workers, using overtime and dedicated reserve workers. In a case study, a new roster for the employees of a small plant is created. Spring and summer seasons are considered separately, assigning a higher amount of vacations for summer. Previously, the plant had 18 employees, high overtime costs and 7 % of sick leaves. It was found out that 2 new employees were needed in order to take absences fully into account. When the new rosters with 20 workers were adopted, overtime costs and the amount of sick leaves were lowered considerably.

Vacations are considered in the scheduling model of Azmat & Widmer (2004), who present the first approach introducing the notion of holiday weeks within annualized hours planning. Constraints for annual work hours and the number of annual holiday weeks are included. The authors present a three step algorithm for solving the problem. The first step is to calculate a minimal workforce size. After that, the need of overtime is estimated by taking into account the weekly demand and each worker's amount of annual holiday weeks. A quota for holidays is assigned for each week so that the need of overtime is minimized. The final step is to allocate holidays and work days to workers. A manager assigns the holiday weeks to employees according to their wishes while respecting the holiday quotas determined earlier. Work days are allocated to workers who are not on holiday, and the workload is balanced between workers. The algorithm produces feasible solutions, but these solutions can be improved by considering optimization criteria such as increasing the amount of consecutive holiday weeks.

The model is developed further by Azmat et al. (2004) by considering a set of Swiss legal constraints. Constraints related to the vacations include that the employees must receive four holiday weeks per year out of which at least two must be consecutive, and workers wishes must be considered when fixing the vacation periods. Objectives of workforce scheduling include determining a minimal workforce to satisfy the weekly demand, minimizing the weekly overtime hours, minimizing the annual overtime hours, balancing the workload among the employees, and generating a workforce schedule for a whole year. The authors use the approaches of Azmat and Widmer (2004) for the first two objectives, and propose an MIP model for fulfilling the remaining three objectives. Two options for assigning holiday weeks are presented, one where employees choose their holidays from a set of possible holiday weeks, and another where an MIP model assigns holiday weeks. In total, 4 different MIP models, 2 for each option of assigning holidays were evaluated using 20 test problems. Differences between the solutions of the two strategies were minor. When the MIP model was used to choose holiday weeks, the solutions included less variation in the working hours assigned to different workers in small problems, but more variation was observed in large problems. Differences in overtime hours were negligible between all the models. In most of the cases, an optimal solution could not be found, but the solutions obtained were feasible and gave a balance of the working hours among the employees which was satisfactory enough in order to make the solutions implementable.

2.3. Days-off optimization

Days-off optimization is closely related to vacation scheduling. In both problems periods without work are assigned to employees. In days-off optimization, single days without work are assigned to a work roster, which usually covers 2-4 weeks, whereas in vacation scheduling weeks without work are assigned to a yearly (or longer) plan. In a number of studies there is a demand that must be satisfied, and a minimum number of workers needed to fulfill the demand is calculated. In other types of approaches vacation- or days-off schedules are created for workers, with the objective function minimizing the sum of multiple worker preference based soft constraint violations. The vacation planning model developed in this thesis differs from most of the vacation planning and days-off optimization approaches in the literature, as it has an objective function of maximizing the smallest amount of surplus workers after assigning vacations (i.e. reserve after all the forecasted work and vacations are assigned to workers). In the model presented in this thesis, the employees cannot be hired or fired, so minimum workforce size is not relevant. Hard constraints are used to enforce solutions to be in accordance with the worker preferences.

One of the earliest models for a days-off optimization problem is presented in Baker and Magazine (1977). The authors present a solution method for a days-off optimization problem where weekdays require N workers and weekend days require n workers. A closed form expression for the minimum number of workers needed as a function of daily demand of workers is also derived. Burns (1978) presents a formula for calculating the optimal workforce size (the minimum amount of workers that can perform the required work). The problem can have differing daily demand for each day of the week. Constraints impose that each worker must have 10 workdays in a 14-day period, each worker must have at least every second weekend off, and a worker cannot have more than six consecutive working days. Burns' formulation is the first one that works for problems where workers must have some weekends off. Bartholdi et al. (1980) use constrained network flow problems for solving cyclic days-off optimization problems. The authors also present a rounding algorithm which enables getting an integer solution from a linear relaxation solution of the problem. Morris and Showalter (1983) model a days-off optimization problem as a set covering problem which is then solved by using a cutting plane method.

Multiple papers deal with days off optimization in presence of hierarchical worker categories where workers with higher qualifications can substitute lower qualified worker. Emmons and Burns (1991) present heuristic algorithms for days-off scheduling of a hierarchical workforce in a case where the demand of workforce is constant. Narasimhan (1997) develops an algorithm which can produce optimal solutions for a single shift scheduling problem of hierarchical workforce. In this problem, work demand is different for weekdays and weekends. Each employee must have two days off every week, every A out of B weekends off, and 5 maximum consecutive working days. Hung (1994) extends the problem by adding variable demand. Workers must have at least n days off in a week ($n \epsilon \{2,3,4\}$). The objective is to find a schedule which has the lowest labor cost and satisfies labor and day off requirements. The author derives necessary and sufficient conditions for feasibility of hierarchical workforce shifts. He also presents an algorithm for generating feasible schedules and proposes a one-pass method for calculating an optimal labor mix. Billionet (1999) presents a similar problem as in Hung (1994), but uses integer programming methods for solving the problem. Optimal solutions can usually be found in a short time. He then extends the model by maximizing the amount of consecutive days off. Binary indicator variables and two constraints on them are used to ensure maximum consecutive days off in the solution.

One way to make scheduling more flexible is to use annualized hours. An annualized hours agreement includes a yearly quota of working hours for each worker, and overtime is given only if the yearly quota is exceeded. This makes it possible to schedule more work in busy periods without the need of overtime hours or hiring temporary workers. As a disadvantage, the use of annualized hours impairs employees' working conditions by making working time irregular and also complicates the planning of their working time. In the model by Hung (2009), annualized hours are used to meet fluctuating demand. Working days and days off are scheduled for the whole year in order to ensure an even working hours distribution among the workers. Holidays are not considered. Corominas et al. (2007) develop an MILP model for solving annualized hours scheduling problems. An example case in a production plant shows that a company's profit can be increased when annualized hours are introduced and the annual working time is reduced to compensate for the irregularity of work. Workers have a period of 6 weeks of holiday annually, divided into one 2 week and one 4 week period. All the workers get holidays at the same time and the plant is shut down during the holiday weeks. Annualized hours agreements allow creating inventory just before the holiday weeks, and this reduces lost demand even though the total yearly working hours are reduced. Hertz et al. (2010) use also an MILP model for annualized hours planning. The workers have 3 holiday weeks annually, and two of them must be consecutive. The authors introduce dummy variables to impose consecutiveness constraint, and this leads to the need of multiple additional constraints. Four different objective functions are presented, but a multi-criteria model is not considered. Instead, the created model can act as a decision management tool by offering multiple objectives to choose from.

Beaumont (1997) describes a days-off optimization problem, in which the amount of work for a whole year is considered for a service company. The problem is formulated as a MILP. The focus in building the model is on long-term solution. Short changes in the amount of workers needed (caused for example by weather and sick leaves) are taken into account by hiring external contract workers or by using workers who agree to do overtime. Cycles of work days and days off must fulfill the contracts made by the workers and management of the company:

- There must be 4-7 consecutive working days.
- During a calendar week there can be at most 5 working days.
- There must be at least 2 and at most 4 consecutive days off.
- There must be 223 or 224 working days during a 47 week cycle.

A cutting plane method was used for solving the problem, and the computation was made with the commercial solver CPLEX. The new approach for creating rosters makes it possible to create different cycles much faster and easier compared to the previous methods used.

Carter and Lapierre (2001) consider a scheduling problem for emergency room physicians, which is considered one of the most challenging physician scheduling problems. Emergency rooms are open 24 hours in each day of the week. The authors present two case studies in which they developed rosters for two hospitals in Montreal.

One of these, Charles-Lemoyne Hospital, used a manually made cyclic roster, which was identical for every physician and included whole weekends off only every third weekend. A new automated rostering method was developed by using Tabu Search metaheuristics. In the new roster physicians had every second weekend off, but not as regularly as before. Physicians' satisfaction for their rosters was improved after implementing the new rosters. The Jewish General Hospital used a roster which was made in many phases, partly manually, partly using computers. The roster was not cyclic. It took 40 hours of work to make a roster for three months. In addition it took 6 weeks to gather all the required information, because physicians go to work irregularly and personal input of vacation and day-off wishes from each physician was needed. The authors modified a certain previously used algorithm to take into account the needs of the hospital, and they got positive feedback from the physician, who was in charge of making the rosters. The model for the Jewish General Hospital was not completed, but the authors expected that the schedules created with it would be better than the old ones.

Beaulieu et al. (2000) formulate an optimization model for scheduling emergency room physicians. Cyclic scheduling methods (scheduling shifts and days-off in repeating cycles) are not applicable for scheduling emergency room physicians because there are too many rules and constraints. Non-cyclic methods must be used instead and those can be based on human expertise and the use of spreadsheets or optimization approaches. Advantages of optimization approaches include the fact that little human intervention is needed and more rules can be handled simultaneously. On the downside, developing an optimization model can take years. With the presented optimization model, results could be obtained in less than one day (human experts needed 5 days in this application) and the solution quality was also improved as fewer soft constraint violations were made.

Costa et al. (2006) also present a non-cyclic problem. Demand is fluctuating and the number of total work days for each employee is fixed. Employees must have 2-3 consecutive days off per week so that in a month there is at least one sequence of three consecutive days off. Multiple constraints are needed to prohibit any other types of days off patterns than two or three consecutive days off. The authors present a three step algorithm that is based on discrete tomography and maximum flow search for solving the problem. The algorithm produces solutions in polynomial time.

Day and Ryan (1997) present a method for rostering flight attendants of short-haul airline operations. Rosters are 14 days long and the problem is non-cyclic. In their approach, days-off allocation and duty allocation problems are solved separately. In the days-off allocation problem all the possible days-off combinations are enumerated (when legal constraints are considered) and a set partitioning problem is solved to obtain a feasible days-off roster. Days off combinations are assigned a base cost representing how much desirable they are. Penalty costs are based on crew preferences. For example a single work day is penalized. In days-off optimization, a single days-off combination is chosen for each crew member so that every day has enough available employees to handle the daily demand. The method was applied to design rosters for flight attendants of Air New Zealand and the solutions found were superior to the manually generated ones in both fairness and days-off distribution.

Knust and Schumacher (2011) develop a model for scheduling tank trucks and truck drivers of a small oil company. The drivers of the company have different skills and there are many types of trucks. The goal of the model is to assign a driver to each duty while taking into account safety and legal regulations. Additional goals include to even out the working time among different drivers and to respect their wishes for vacation times. The model is formulated as a MIP which includes 8 hard constraints and 4 soft constraints. The objective function is to minimize the weighted sum of some soft constraint violations. The model is so complicated that optimal solutions cannot be calculated without using an expensive commercial optimization software, and the company does not want to buy one. Consequently, the authors decided to use a two phase solution process. First a reduced MIP is solved, where the two least important soft constraints are removed. The second phase is to improve the solution with respect to the previously ignored soft constraints. Good solutions are obtained already with computing times of about 10 minutes. The commercial optimization software CPLEX could solve the whole problem, and yielded better results, but the computing times were several hours. The developed method for scheduling was considered a good improvement over the previous manual planning process, as feasible solutions were hard to find and there were significant differences in the amount of work assigned to different drivers.

Multiple other types of modeling and solution methods for days-off optimization problems are also presented in the literature. Al-Zubaidi and Christer (1997) use simulation models to evaluate different maintenance management policies of maintenance work for a hospital building. Emmons and Fuh (1997) derive formulas for calculating minimal cost full time workforce when part time workers are available. Also, a constructive algorithm for generating schedules is presented. Full time workers must have 2 days off in a week and every *A* out of *B* weekends off. Bellanti et al. (2004) present a greedy-based neighborhood search approach for scheduling nurses in which days off and shifts are scheduled simultaneously. The algorithm also considers holidays and requested days off. Nurmi et al. (2011) utilize a population-based local search method with greedy hill-climbing mutation for scheduling bus drivers. A real world application for a Finnish transit company is presented. Days-off scheduling and shift scheduling are considered separately, and the objective of days-off scheduling is to minimize the weighted sum of some soft constraint violations. The weights of the soft constraints represent their importance. The company was very satisfied with the results produced by the algorithm. The time needed for creating rosters was reduced, shifts and days off were more balanced among drivers, and drivers had less idle time than before.

2.4. Scheduling with constraint programming methods

Constraint programming methods have been used in various scheduling applications. They are especially useful for highly constrained problems, as they allow handling complex and diverse constraints. Constraint programming is also very effective for finding feasible solutions to constraint satisfaction problems (He and Qu 2012).

Laborie and Rogerie (2008) present a framework for scheduling problems, which is based on a new type of variables called *conditional time-interval variables*. Conditional intervals represent tasks which may be included in a final schedule, but are not mandatory. These are important in many applications. In constraint-based scheduling, conditional time-intervals are usually modeled by imposing global constraints over ordinary integer variables. Time-interval variables embed conditionality intrinsically, which simplifies modeling. The authors also present various constraints that can be imposed on the time-interval variables. Later, Laborie and Rogerie (2009) introduce additional concepts for modeling scheduling problems with the conditional time-interval framework. For example *cumul function* expressions can be used in scheduling problems that involve usage of cumulative resources to sum the contributions of individual interval variables. Detailed description about conditional interval variables, cumul functions and their usage on the problem of vacation planning is presented in chapter 3.3.

Constraint programming has been used in solving various nurse rostering problems. Cheng et al. (1996 and 1997) use constraint programming with a redundant modeling approach, where semantically redundant constraints are used to increase the propagation and pruning of original constraints. Nurse Rostering is modeled as a constraint satisfaction problem, and constraint programming tools are used to build up a constraint solving engine. Soft constraints are modeled as choices in solution search tree. Two different models are created, one with nurses as variables and shifts as their domain and another one with shifts as variables and nurses as their domain. Authors use redundant modeling for pruning and propagation of constraints between sub models. This speeds up the solution process considerably. Models are connected with channeling constraints which express relationship among variables in different models.

The authors state that there are three main approaches for modeling nurse rostering problems. The first one is to use mathematical modeling and to apply operations research methods. It provides relatively fast solution times, but it is rigid. The second option is to design problem-specific heuristics. These are not generic and successful algorithms for one problem do not necessarily perform well for different problems. The third approach is to use constraint programming. It is flexible as small changes in the problem can be implemented with small changes in the model, and has great expressiveness. On the downside, representation of variables and constraints might not be easy. The authors highlight that soft constraints are often mutually incompatible. The constraint programming methodology allows a high level of flexibility in changing the problem specifications which is difficult to achieve with the two other approaches.

He and Qu (2012) present a constraint programming based column generation (CP-GG) solution procedure for solving complex nurse rostering problems. A simple heuristic is used to get initial solutions instead of using specific propagation algorithms. First, all soft constraints are treated as hard, and then relaxed one by one until a feasible solution is found. At each iteration of the column generation algorithm, the pricing sub-problem is solved to find columns with negative reduced cost. The column with the smallest reduced cost is not necessarily the column that causes the largest decrease in the objective function. Actually, any feasible column with negative reduced cost is a candidate to enter master problem. It is easier to generate feasible rather than optimal columns, but then the convergence to the optimum can be slower. The pricing problem is solved by constraint programming, and also columns with reduced cost larger than the minimum are added to the master. However, all columns with cost larger than a predefined upper bound are ignored. Also Depth Bounded Discrepancy Search (Walsh 1997) is used to improve the efficiency of CP. The proposed method is competitive compared to previous approaches although it does not include any metaheuristics.

Constraint programming has also been used in generating solutions for crew scheduling problems. Fahle et al. (2002) present a CP-GG approach for solving airline crew rostering problems. The authors state that usually these problems are divided into sub-problem, which is used to generate legal rosters, and a master problem, where the generated rosters are assigned to crew members. The airline regulations considered in this study are however so complex that previous approaches cannot be used to model all of them. To overcome this difficulty, the authors formulate the sub-problem as a constraint satisfaction problem which allows modeling all the constraints. The proposed method is tested with real data.

Sellman et al. (2002) present two algorithms for solving airline crew rostering problems: the CP-GG algorithm presented by Fahle et al. (2002) and a constraint programming based heuristic tree search. The authors describe strengths and weaknesses of both algorithms and test their performance. Finally, the algorithms are combined to overcome their intrinsic limitations. Results of test cases show that the hybrid algorithm is superior compared to the two original algorithms.

Hybrid approaches are also used by Yunes et al. (2005). They deal with a crew management problem of a bus company that operates in Belo Horizonte, Brazil. The crew management problem is split into two sub-problems, a crew scheduling problem and a crew rostering problem. Mathematical programming and constraint logic programming are applied for solving each problem, and finally hybrid column generation approaches that combine both methods are developed. Examples with real world data are used to test the different algorithms. Hybrid approaches were the best performers producing high-quality solutions with reasonable computational times.

3. Vacation optimization model

3.1. Background

Finland's train drivers go to work from about 25 depots. Each driver starts working from his/her home depot. Drivers cannot be transferred between depots but some of the workload can be transferred through duty re-planning if needed. On average, 650 freight and long distance trains are driven each day. In many cases the traffic is seasonal, so there can be considerable changes in the amount of trains driven, especially on a single depot level. Drivers must also participate in various trainings to maintain their license to drive. The frequency of different types of trainings varies from 1 to 5 years. In addition, the drivers have licenses for different types of locomotives; not all drivers can drive all the train types that start from his/her home depot.

Each driver is granted a certain amount of vacations for each calendar year; law and contract based rules set the constraints on how the vacations must be planned. A calendar year is split into three vacation seasons: spring (January-May), summer (June-September) and fall (October-December). Most of the drivers have 5 weeks of "ordinary" vacation in a calendar year, but experienced drivers have a larger vacation allowance of 6 weeks. Drivers earn extra vacation days for the fall if they do not have all the possible summer vacation days in the summer vacation season. When extra days are taken into account, there are 3 main possibilities for dividing the vacation allowance of a driver inside one year. Short vacation drivers can have (including extra days, spring, summer and fall) either $12 + 15 + 9 = 36$ days or $9 + 18 + 6 = 33$ days of vacation, and drivers with long vacation allowance usually have $15 + 18 + 12 = 45$ days of vacation. In vacation modeling, weeks are split into two parts (*half weeks*): Monday-Wednesday and Thursday-Sunday. Each such half week consumes three vacation days while Sundays and public holidays do not consume vacation days.

Drivers' rosters are planned for 3 week planning periods; a driver has a maximum working time of 114 hours and 45 minutes in 3 weeks. However, if a planning period includes public holidays, the maximum working time limit is lowered by 8 hours for each such public holiday. This means that in a planning period that includes public holidays more drivers are needed to perform the same amount of work as in a period without public holidays. The public holidays in Finland that have the greatest impact on vacation planning process are Christmas, Easter and Midsummer. If a driver's workload exceeds the maximum working time, the driver gets overtime, which is expensive for the train operator.

In addition to normal work, drivers can be in training, on vacation, or absent for other reasons. Absent drivers are mostly on sick leave, and the share of absent drivers is nearly constant through the whole year. Trainings reduce the amount of normal work that a single driver can do in a planning period. For example, if the maximum working time is 114 hours and 45 minutes and a driver has 16 hours of training, he/she can work on driving duties for 98 hours and 45 minutes. Each vacation day that a driver has in a certain planning period reduces the driver's maximum working time in that period by $\frac{114.75}{21}$ hours (about 5 hours and 28 minutes), where 21 is the number of days in a 3 week planning period and 114.75 are the maximum working hours of the planning period.

Until late 2014, drivers' vacations were planned manually using Excel. The amount of drivers allowed on vacation was forecasted only at the vacation season level while in reality, for example, the planning period that includes the holidays of Christmas allows giving a very limited amount of vacations before overtime work is needed. As plans were made by hand, making even a small change in the plans, for example, to fulfill a wish of a single driver resulted often in a lengthy operation. Often, when one vacation is moved, multiple other vacations must also be shifted slightly forward or backward to balance the amount of vacations and to ensure sufficient workforce at work. As a result of these challenges, an optimization model for vacation planning was created and it was used in planning the drivers' vacations for the year 2016.

In the new vacation planning process, forecasts for the amount of work and trainings are made at planning period level for each depot. As the forecasts are made at a more accurate level than before, gathering the required information is more difficult. On the other hand, the increased level of detail makes it possible to limit the amount of allowed vacations in the planning periods which include public holidays or a large amount of trainings. After the working hour forecasts are made, the amount of forecasted work for each planning period is divided by the amount of maximum working hours in the corresponding planning period. In this way, base level information about the number of workers needed for each planning period is obtained. The number of workers available in each half week period must also be determined. Information about retirements, new recruitments and fixed absences (such as maternity/paternity leave) is gathered, and their effect on the number of available workers is calculated. Next, the forecasted amount of workers needed at work is subtracted from the number of workers available in the personnel base. The result is the number of reserve workers available after all the forecasted work is assigned to the workers. The amount of reserve workers in each period sets an upper bound for the amount of possible vacations in that period.

When the amount of reserve for each half week is obtained, the information is entered into the optimization model. The goal of the vacation optimization is to maximize the minimum reserve (i.e., the amount of surplus workers) for each depot and for each vacation season after all the vacations are assigned to the workers. This forces the reserve of the personnel base to be split as evenly as possible over all the half-weeks of the vacation season, and leaves as much as possible room for sick leaves, other absences and unexpected changes in demand and training needs.

As an example, consider a base with 52 drivers. If the forecasted amount of work for a planning period is 5000 hours and the maximum working time is 114 hours and 45 minutes, the amount of drivers needed to perform the work is $\frac{5000}{114.75} = 43.57$, so 44 drivers would be needed. If maximum working time for another period with 5000 hours of work is 106 hours and 45 minutes, the same calculation gives $\frac{5000}{106.75} = 46.83$, so 47 drivers are needed. Thus, the period with 106 hours and 45 minutes of maximum working time needs 3 more drivers at work. This means that if the optimal reserve is 4 drivers, $52 - 44 - 4 = 4$ drivers can be on vacation at the same time in the first planning period but only 1 driver can be on vacation at the same time during second planning period.

A comparison of expected (forecasted) results of vacation planning between the former planning method and the new vacation planning optimization model is presented in Figures 1 and 2 for a time period of 17 planning periods, which is close to one year. The first 6 periods are in the summer vacation season, where the demand for trains is smaller than in the rest of the year and drivers must have long vacations. The planning periods 7-11 belong to the fall vacation season and the planning periods 12-17 to the spring vacation season. The black line represents the number of drivers which decreases in time due to retiring drivers, but has one jump upwards as new drivers are recruited. The bars represent how many drivers are needed in each planning period, with a legend describing the maximum working hours in each planning period. A driver can be driving trains, in training, on vacation or absent for other reasons. The numbers in the bars present the average amount of drivers on vacation in each planning period.

The former model for allocating vacations (Figure 1) uses equally sized vacation groups. In the summer there are about 160 drivers on vacation, in the fall about 103 drivers on vacation and in the spring about 108 drivers on vacation. The only thing that influences the amount of drivers on vacation in addition to the change of vacation season is the retirement of drivers, which causes a slow decrease in the amount of vacations. There are 5 planning periods where the number of drivers needed is at least 20 bigger than the number of drivers available. In the worst case, there is a shortage of 52 drivers (over 6 % of the available workforce). This causes a need for costly overtime work which raises the company's personnel costs considerably.

The number of drivers needed for driving trains and trainings is fixed for each planning period, and the buffer for other absences is a fixed percentage of the drivers. Thus, the only number that can be changed is the number of drivers on vacation. The new vacation planning model was created with the goals of making vacation allocation more cost-efficient and less time-consuming. The expected results of the new model (Figure 2) include no planning periods with significant need for overtime. There are some planning periods with greater number of drivers needed than available, but the maximum shortage is only 7 drivers, which is under 1 % of the available workforce. Now the amount of drivers on vacation varies significantly between and within the vacation seasons, and the vacations are used to make the number of workers needed in each planning period to be as close as possible to the amount of workers available. With these vacation plans, the company would have only small needs for overtime, and could reduce personnel costs significantly.

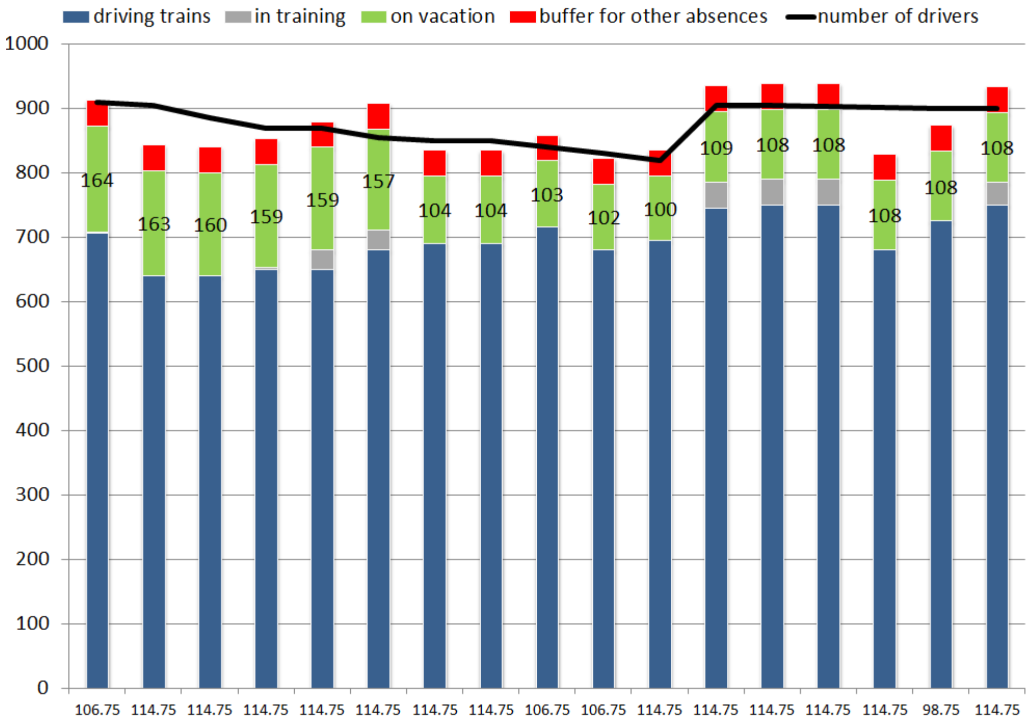

Figure 1: Expected results of the former vacation planning model

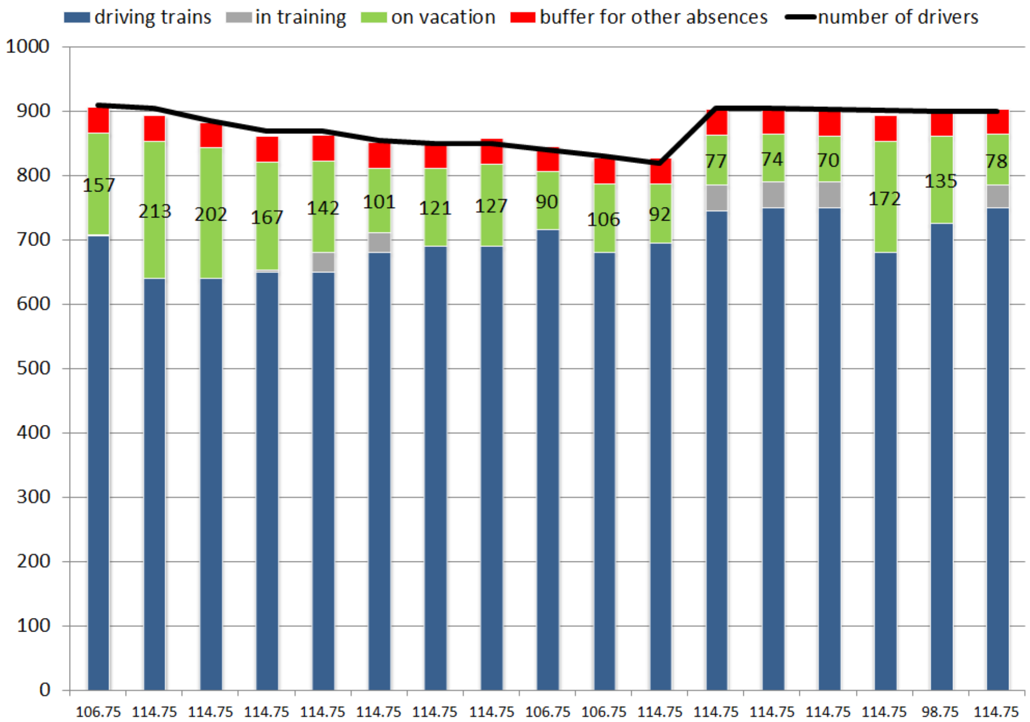

Figure 2: Expected results of the new vacation planning model

3.2. Problem description

This chapter describes the vacation planning problem at VR. The problem is described for a single personnel base and a single vacation season. The personnel base has workers $m = 1,2, \ldots, M$ and the vacation season contains half weeks $n = 1,2, \ldots, N$. These half weeks (Monday-Wednesday or Thursday-Sunday) are referred to by *periods*. A worker is either on vacation or at work. Special cases such as workers who are retiring are not included in the optimization problem, but are taken into account in base data for the problem. The problem's base data includes maximum amount of workers allowed on vacation p_n in each period n. Also, previous year's vacation start period for each worker l_m, ending period of vacation in the previous vacation season for each worker v_m and vacation allowance in days a_m ($\frac{a_m}{3} \in \mathbb{N}$) for each worker are known. Sundays and public holidays do not consume vacation days.

The base data includes information about special skills $j = 1,2, \ldots, J$ for each worker. These are represented with:

$$s_{m,j} = \begin{cases} 1, \text{ if worker } m \text{ has special skill } j \\ 0, \text{ otherwise} \end{cases}.$$

The maximum for workers with skill j on vacation at the same time $s_{max,j}$, $j = 1,2, \ldots, J$ is also given. Finally, the base data includes information about forbidden vacation start periods (vacations cannot start on public holiday periods and second periods of each planning period):

$$h(n) = \begin{cases} 1, \text{ if vacations cannot start at period } n \\ 0, \text{ otherwise} \end{cases}.$$

The following constraints must be satisfied, when the vacations are planned:

(C1) Each worker has exactly one continuous vacation in the vacation season.

(C2) In the basic case, the length of the vacation is the same as vacation allowance. If the vacation overlaps multiple public holidays, the length of the vacation is increased by one period, i.e. 3 days (excluding the public holidays that are on Sunday, because Sundays already do not consume vacation days).

(C3) Each worker's vacation must start at least f periods after the previous year's vacation start period in the same vacation season.

(C4) If worker A's previous year's vacation has started before worker B's previous year's vacation, worker A's vacation must start before or at the same time as worker B's vacation. The vacations for workers that had previous year's vacation close to the end of the vacation season must "jump" to the beginning of vacation season because of the previous constraint. The start periods of these vacations that jump to an earlier period $\bar{n}$ because of the previous constraint are modeled as $\bar{n} + N$ in order to satisfy constraint (C4).

(C5) The amount of workers on vacation in each period is less than or equal to the maximum amount of workers allowed on vacation p_n.

(C6) The amount of surplus workers in each period ($p_n -$ the amount of workers on vacation in period n) must be equal to or greater than w_{min}.

(C7) The amount of surplus workers in each period must be equal to or smaller than w_{max}.

(C8) The maximum amount of workers with skill j allowed on vacation at the same time is $s_{max,j}$.

(C9) A vacation cannot start in the second period of a planning period (period which starts on the first Thursday of a 3-week planning period) as it would leave only 3 working days in the beginning of planning period, making the work of roster planning personnel difficult.

(C10) Vacations cannot start on public holiday periods.

(C11) Vacations cannot start so late that the ending point of the vacation belongs to the next vacation season

(C12) Vacations must start at least g periods after the ending period of vacation in the previous vacation season. This difference of g periods is referred to as the minimum gap between vacations.

There is also one additional constraint that is not always necessary, but which is useful for improving solution quality in some problems, where vacations tend to shift forward too much.

(C13) The maximum number of periods that the vacation of the first driver can shift forward is $f + f_{max}$, i.e. the shift forward must be in the range $[f,\ f + f_{max}]$.

For the purpose of imposing constraints (C3), (C4) and (C12) the set of the N periods is viewed circularly so that, for example the fifth period after od $N - 2$ is considered to be period 3. In order to simplify the model, we use N

additional periods $N+1, N+2, \dots, 2N$ to model vacation starts that are moved to an earlier period with respect to the previous vacation season due to constraints (C3), (C4) and (C12). We will also informally say that such vacations "jump" to the beginning of the vacation season.

The objective of the vacation planning is to maximize the minimum surplus workers w_{min} over all periods. This objective ensures that a sufficient amount of workers are available in every period, leaving room for absences such as sick leaves. In addition, minimizing the maximum surplus workers w_{max} can be added to the objective in order to even out the distribution of surplus workers as much as possible. For example if there are 10 periods and the optimal w_{min} would be 3, the surplus workers for each period could be $[3,3,3,3,3,3,3,3,3,10]$. By adding the second criteria to the objective the solution could change to $[3,3,3,4,4,4,4,4,4,4]$ (if constraints allow it).

The first approach that was tested for solving the vacation planning problem was to formulate it as an MILP by defining a binary variable for each period of each driver (taking value 0 if the period is a working period and value 1 if the period is a vacation period). This model proved to be inefficient in tests with real world data. Especially the consecutiveness requirement of vacations increased the amount of constraint needed radically, making running times long if optimal solution even could be found.

A constraint programming approach was tested next and it proved to be much more efficient compared to the MILP. Optimal solutions were found quickly, even the largest real world problems could be solved in a few minutes. The approach was implemented for practical use for planning vacations of train drivers for the year 2016. Concepts used in modeling the vacation planning problem with constraint programming are presented in Chapter 3.3. and a constraint programming formulation of the problem is presented in Chapter 3.4.

3.3. Modeling with constraint programming

Laborie and Rogerie (2008 and 2009) present concepts that are useful in presenting a constraint programming formulation of the vacation planning problem. The authors introduce a new type of variables, called *conditional interval variables*. These are very useful in modeling vacations as they allow expressing each vacation period with a single variable, which has an innate characteristic of consecutiveness. Another concept that is used in the formulation is that of *cumul functions,* which allow summing the conditional interval variables. The notation used in this thesis is adapted to practical use in vacation planning. For a more formal description of the concepts presented in this thesis the reader is referred to Laborie and Rogerie (2008 and 2009).

In the case of vacation planning, the vacation of each driver m is represented by a single decision variable x_m, which is an interval variable. The domain of the variable is $[b, N+1) | b, N \in \mathbb{Z},\ b \le N+1$, where b is the first period of vacation season being planned and N is the last period of vacation season being planned. The upper bound for the variable value is $N+1$ because if a vacation includes period N, it ends just before period $N+1$ starts. Each interval variable x_m has an integer start time b_m, an integer end time e_m and a non-negative integer duration $z_m = e_m - b_m$.

As an example, three vacations are presented as interval variables in Figure 3. The domain of the vacations is now [1,19) and driver A has a vacation x_A with start time $b_A = 3$, end time $e_A = 6$ and duration $z_A = 3$ periods. Vacations are marked with green color and working periods with red color.

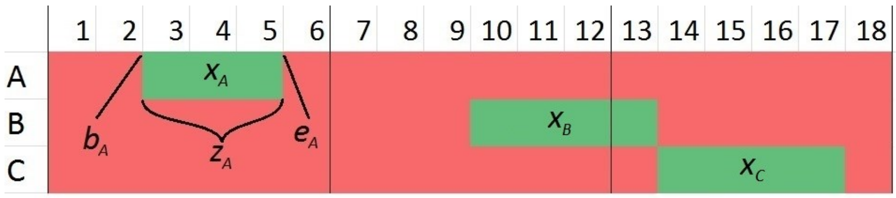

Figure 3: Vacations as interval variables

Drivers' vacation lengths are predetermined, but as public holidays do not consume vacation days, some vacations must be extended by 1 period. For this reason, conditional interval variables are used. Conditional interval variables have one additional characteristic compared to interval variables, namely, an execution status r_m. If a conditional interval variable X_m is executed, it is said to be present ($r_m = 1$) and it has a start time time b_m, an end time e_m and a duration z_m. If a conditional interval is not executed, it is said to be absent ($r_m = 0$) and it does not have any start time, end time or duration. Absent interval variables are not considered by constraints. To deal with the possibility of vacation extension due to public holidays, each driver m is assigned two conditional interval variables $X_{m,k}$ $(k = 0,1)$ with start time time b_m, end time $e_{m,k}$, duration $z_{m,k}$ and presence $r_{m,k}$ ($r_{m,k} = 1$, if the corresponding interval variable is present and $r_{m,k} = 0$, if the corresponding interval variable is absent). The value of k indicates how many vacation periods are added to the length of a driver's vacation in addition to the driver's vacation allowance. Both possible vacations start at the same time, but $X_{m,1}$ has a one period longer duration. One of the variables $X_{m,k}$ for each driver is always present and the other one is absent. The variable $X_{m,k}$ that is present is the vacation that the driver gets, and the other one is ignored. The vacation that is actually assigned to driver m is modeled by using an interval variable x_m ($x_m = X_{m,0}$ or $x_m = X_{m,1}$) which was presented in previous paragraph. As all variables x_m are present, those can be modeled as interval variables without conditionality.

The choice between conditional intervals $X_{m,0}$ and $X_{m,1}$ is illustrated in Figure 4. In this example the driver has a vacation allowance of 2 periods, and period 5 has public holidays, so if vacation overlaps it, the vacation is extended by 1 period. If the vacation of the driver starts at period 3 or earlier, $x_m = X_{m,0}$, because the vacation ends before the public holiday period. If the vacation of the driver starts at period 4, $x_m = X_{m,1}$, because the second period of the vacation $X_{m,0}$ overlaps the public holiday period, which does not consume vacation days. The vacation cannot start at period 5, because of constraint (C11). If the vacation starts at period 6 or later, $x_m = X_{m,0}$, because the vacation does not overlap the public holiday period.

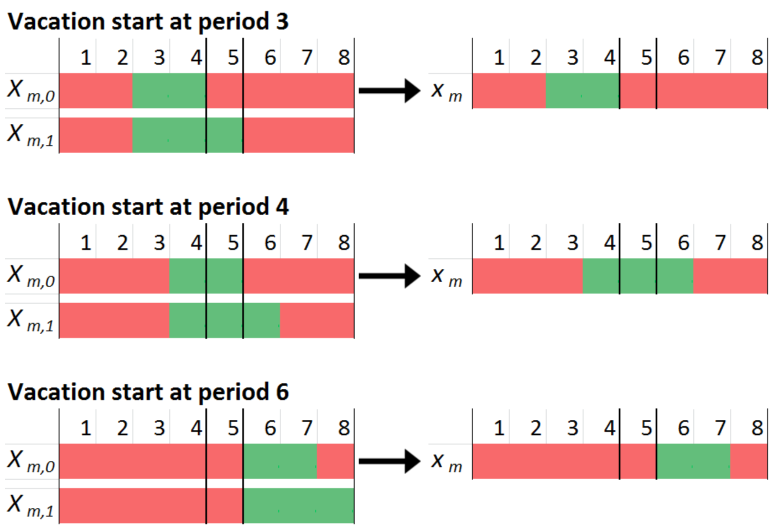

Figure 4: Choice of $\boldsymbol{x_m}$ illustrated for a vacation allowance of two periods

Laborie and Rogerie (2009) also present the notion of *cumul function* to enable modeling cumulated usage of resources, which are produced and consumed by activities. Individual activities can be modeled as interval variables or fixed intervals of time. Elementary cumul functions are used to describe the contributions of individual activities. The usage of a cumulative resource can be modeled with *pulse* function and resource production/consumption can be modeled with *step* function. The elementary cumul functions are presented in Figure 5, where a is an interval variable, and $[u, v]$ is a fixed interval of time and h is the height of the step. The two functions on the top row are defined with the fixed time interval, and the interval variable is used to define the rest of the functions.

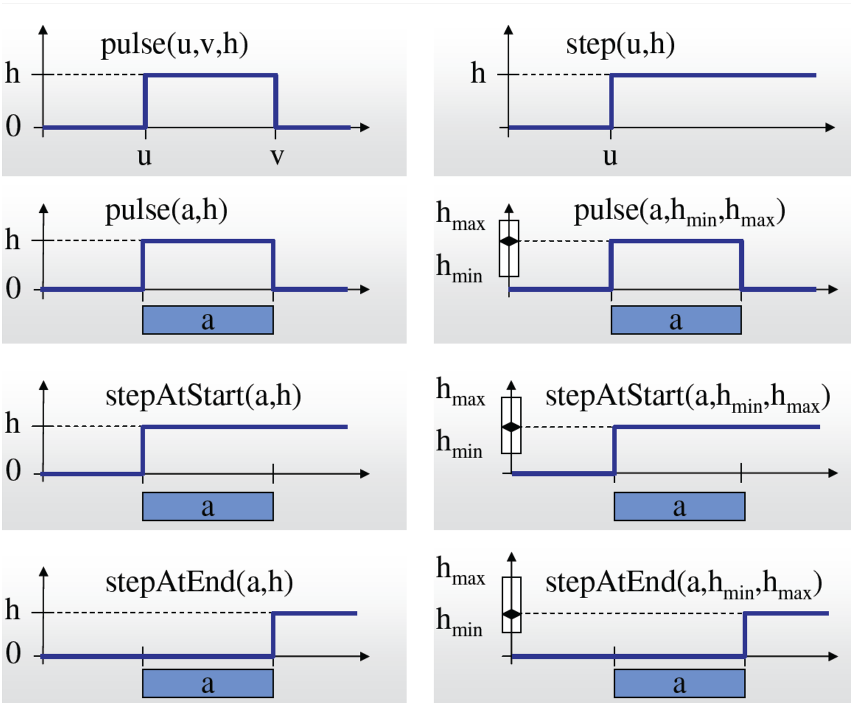

Figure 5: Elementary cumul functions (Laborie and Rogerie 2009)

In the vacation planning model the starting points of vacations and their usage of reserve are modeled with *stepAtStart* functions. Those can be simplified to *step* functions when resource consumption h is set to be 1 for each vacation and the point u where the reserve is consumed is set to be b_m. Similarly, the ending points of the vacations are modeled with *stepAtEnd* functions which can be simplified to step functions by setting $u = e_m$ and $h = 1$. Figure 6 illustrates how these step functions can be used to obtain the total amount of drivers on vacation on each period. In the upper figure, red color cells with entry 0 indicate a working period and green color cells with entry 1 indicate a vacation period. The figure labeled vacations started shows the step function modeling the sum of started vacations and the figure labeled vacations ended shows the step function modeling the sum of ended vacations. The number of ongoing vacations is calculated by subtracting the vacations ended from the vacations started, and the corresponding step function is shown in the figure "ongoing vacations".

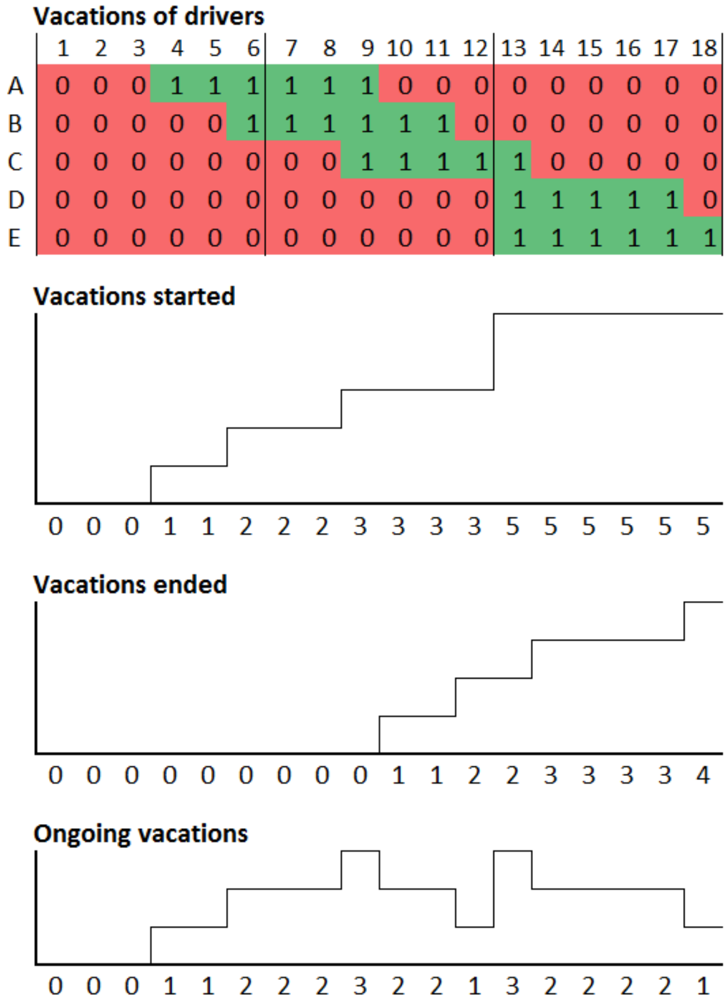

Figure 6: Calculating the amount of ongoing vacations

The amount of ongoing vacations in each period can be subtracted from the maximum amount of allowed vacations p_n to obtain a function which describes the surplus workers (reserve) in each period. Now w_{min} can be set to be the lower bound for surplus workers and w_{max} can be set to be the upper bound for surplus workers to enable modeling the required objective function.

3.4. Constraint programming formulation of the vacation planning problem

3.4.1. Decision variables

Decision variables $X_{m,k}$ $(m = 1,2, \dots, M, k = 0,1)$ are conditional interval variables (see Chapter 3.3.). Each variable represents a single possible vacation. The conditional interval variables are characterized by a start value b_m, an end value $e_{m,k}$, a duration $z_{m,k}$, and a presence $r_{m,k}$($r_{m,k} = 1$, if the corresponding interval variable is executed and $r_{m,k} = 0$, if the corresponding interval variable is absent). Parameter k indicates how many periods are added to the vacation on top of the vacation allowance. Each driver's actual vacation is represented as an interval variable x_m ($x_m = X_{m,0}$ or $x_m = X_{m,1}$), which has start time b_m, end time e_m and duration z_m.

Dummy interval variables d_m are used to model cases where $b_m > N$. The dummy interval variables have start value b_{d_m}, end value e_{d_m} and duration z_{d_m}. The dummy interval variables are vacation interval variables x_m shifted backwards by N periods: $b_{d_m} = b_m - N$, $e_{d_m} = e_m - N$ and $z_{d_m} = z_m$. The total amount of vacations in each period $n = 1 \dots N$ is obtained by summing the vacation interval variables and dummy interval variables. Variables modeling the minimum surplus workers w_{min} and maximum surplus workers w_{max} are also included in the model.

3.4.2. Expressions involving decision variables

Step functions are used to count how many vacations have started and ended. First each worker's vacation start time (of the actual vacation x_m) is modeled with the step function

$$V_{start,m}(n) = \begin{cases} 0, & n < b_m \\ 1, & n \geq b_m \end{cases}, \quad m = 1,2, \dots, M. \tag{3.1}$$

Step functions are also used to model the start times of dummy interval variables

$$D_{start,m}(n) = \begin{cases} 0, & n < b_{d_m} \\ 1, & n \geq b_{d_m} \end{cases}, \qquad m = 1,2,\dots,M. \tag{3.2}$$

By summing the step functions for vacation and dummy variables we obtain

$$V_{start}(n) = \sum_{m=1}^{M} V_{start,m}(n) \tag{3.3}$$

$$D_{start}(n) = \sum_{m=1}^{M} D_{start,m}(n) \tag{3.4}$$

The total amount of vacations started in periods $n = 1,\dots,N$ is obtained by summing the step functions for vacation variables and dummy variables.

$$V_{start,total}(n) = V_{start}(n) + D_{start}(n) \tag{3.5}$$

The same process is repeated for the ending of vacations. First each vacation's and dummy interval's ending point is represented with the step functions

$$V_{end,m}(n) = \begin{cases} 0, & n < e_m \\ 1, & n \geq e_m \end{cases}, \qquad m = 1,2,\dots,M, \tag{3.6}$$

$$D_{end,m}(n) = \begin{cases} 0, & n < e_{d_m} \\ 1, & n \geq e_{d_m} \end{cases}, \qquad m = 1,2,\dots,M. \tag{3.7}$$

Then, the ending functions are summed

$$V_{end}(n) = \sum_{m=1}^{M} V_{end,m}(n), \tag{3.8}$$

$$D_{end}(n) = \sum_{m=1}^{M} D_{end,m}(n). \tag{3.9}$$

The total amount of vacations ended in periods $n = 1,\dots,N$ is obtained by summing vacation- and dummy variable step functions

$$V_{end,total}(n) = V_{end}(n) + D_{end}(n). \tag{3.10}$$

Total amount of workers on vacation in periods $n = 1,\dots,N$ is obtained by subtracting the total amount of vacations ended from the total amount of vacations started

$$V_{total}(n) = V_{start,total}(n) - V_{end,total}(n). \tag{3.11}$$

For each special skill j, the total amount of workers with skill j on vacation in each period is obtained in a similar way as the total amount of workers in vacation. The starting points of each vacation and dummy variable are represented with the step functions of equations (3.1) and (3.2). The total amount of workers with skill j who have started vacation is obtained by multiplying each vacations step function with $s_{m,j}$ and summing the multiplied functions

$$V_{start,j}(n) = \sum_{m=1}^{M} s_{m,j} V_{start,m}(n), \qquad j = 1,2,\dots,J, \tag{3.12}$$

$$D_{start,j}(n) = \sum_{m=1}^{M} s_{m,j} D_{start,m}(n), \qquad j = 1,2,\dots,J. \tag{3.13}$$

The total amount of vacations started for workers with skill j is then obtained as

$$V_{start,j,total}(n) = V_{start,j}(n) + D_{start,j}(n), \qquad j = 1,2,\dots,J. \tag{3.14}$$

The same process is repeated for the ending of vacations for workers with special skill j. First each vacation's and dummy interval's ending point is represented with a step function given by equations (3.6) and (3.7). The total amount of workers with skill j who have ended vacation is obtained by multiplying each vacation step function with $s_{m,j}$ and summing the multiplied functions.

$$V_{end,j}(n) = \sum_{m=1}^{M} s_{m,j} V_{end,m}(n), \qquad j = 1,2,\dots,J, \tag{3.15}$$

$$D_{end,j}(n) = \sum_{m=1}^{M} s_{m,j} D_{end,m}(n), \qquad j = 1,2,\dots,J. \tag{3.16}$$

Total amount of vacations ended for workers with skill j is obtained as

$$V_{end,j,total}(n) = V_{end,j}(n) + D_{end,j}(n), \qquad j = 1,2,\dots,J. \tag{3.17}$$

The total amount of workers with skill j on vacation for each period $n = 1 \dots N$ is obtained by subtracting the total amount of vacations ended from the total amount of vacations started

$$V_{total,j}(n) = V_{start,j,total}(n) - V_{end,j,total}(n), \qquad j = 1,2,\dots,J. \tag{3.18}$$

Surplus workers $W(n)$ for periods $n = 1 \dots N$ can now be calculated by subtracting the total vacations from the allowed vacations

$$W(n) = p_n - V_{total}(n). \tag{3.19}$$

3.4.3. Constraints

Each worker's vacation must start at least f periods after the previous year's vacation start period in the same vacation season due to constraint (C3). If a worker's vacation jumps from the end of the vacation season to a period $\bar{n}$ in the beginning of the vacation season because it is shifted forward, then the vacation is given a starting period $N + \bar{n}$. This constraint is modeled as follows

$$b_m \geq l_m + f, \qquad m = 1,2,\dots,M. \tag{3.20}$$

Equation (3.20) ensures that constraint (C3) is fulfilled.

If worker A's previous year's vacation has started before worker B's previous year's vacation, worker A's vacation must start before or at the same time as worker B's vacation. Workers can be sorted by previous year's vacation start date in the base data, so the constraint can be presented in a simple form

$$b_m \geq b_{m-1}, \qquad m = 2,3,\dots,M. \tag{3.21}$$

Another constraint is needed to ensure that the vacation start of the last driver is not after the vacation start of the first driver

$$b_1 \geq b_M - N. \tag{3.22}$$

Equations (3.21) and (3.22) add constraint (C4) to the model.

The maximum number of periods that the vacation of the first driver can shift forward can be set with the constraint

$$b_1 \leq l_1 + f + f_{max}. \tag{3.23}$$

Equation (3.23) is used to add constraint (C13) to the model, when necessary.

The duration of a vacation is usually the same as the vacation allowance, but it must be increased by 1 period, if a vacation overlaps multiple public holiday days. For this reason the durations of possible vacations $X_{m,k}$ are set to be

$$z_{m,k} = a_m + k, \qquad m = 1,2,\dots,M, \qquad k = 0,1. \tag{3.24}$$

Each worker's vacation end period is equal to the start period plus the duration (for example, if a vacation's duration is 3 periods and the starting period is 5, the vacation period will end at the beginning of period 8)

$$e_{m,k} = b_m + z_{m,k}. \tag{3.25}$$

Each worker m has exactly one vacation, so always either $X_{m,0}$ or $X_{m,1}$ is chosen as the actual vacation x_m. This can be represented using the presence characteristic of the conditional interval variables $X_{m,k}$ by imposing the constraint

$$r_{m,0} + r_{m,1} = 1, \qquad m = 1,2, \dots, M. \tag{3.26}$$

With equation (3.26) Constraint (C1) is fulfilled as each driver gets exactly one vacation.

The duration of a vacation of driver m must be extended to be $a_m + 1$, if the vacation overlaps multiple public holidays. When n' is used to mark a period with multiple public holidays, the extension of vacation can be modeled with the constraint

$$r_{m,1} = 1, \qquad b_m = n' - a_m + 1, \dots, n'. \tag{3.27}$$

Equation (3.27) is used to enforce constraint (C2).

Constraints are also used to forbid vacation start at certain periods i.e. public holiday periods and second periods of each planning period (periods starting at the first Thursday of each planning period). If period n is a forbidden vacation starting period, $h(n) = 1$, otherwise $h(n) = 0$. The forbidden vacation starting periods are modeled with

$$b_m \neq nh(n), \qquad n = 1,2, \dots, N. \tag{3.28}$$

Equation (3.28) ensures that constraints (C9) and (C10) are fulfilled.

A vacation also cannot start in a period which would lead the ending point of the vacation to be on the next vacation season

$$e_{m,k} \leq N + 1, \qquad m = 1,2, \dots, M, \qquad \text{if } b_m = 1, \dots, N, \tag{3.29}$$

$$e_{m,k} \leq 2N + 1, \qquad m = 1,2, \dots, M \text{ otherwise}. \tag{3.30}$$

Constraint (C11) is fulfilled with equations (3.29) and (3.30).

The amount of workers on vacation in each period must be less than or equal to the maximum amount of workers allowed on vacation

$$V_{total}(n) \leq p_n, \qquad n = 1, \dots, N. \tag{3.31}$$

Equation (3.31) forces fulfillment of constraint (C5).

The amount of surplus workers in each period must be equal to or greater than w_{min}

$$W(n) \geq w_{min}, \qquad n = 1, \dots, N. \tag{3.32}$$

Equation (3.32) adds constraint (C6) to the model.

The amount of surplus workers in each period must be equal to or smaller than w_{max}

$$W(n) \leq w_{max}, \qquad n = 1, \dots, N. \tag{3.33}$$

Equation (3.33) enforces constraint (C7).

The maximum amount of workers with skill j allowed on vacation at the same time is $s_{max,j}$

$$V_{total,j}(n) \leq s_{max,j}, \qquad n = 1, \dots, N, \qquad j = 1,2, \dots, J. \tag{3.34}$$

Constraint (C8) is fulfilled with equation (3.34).

Finally, constraint (C12) states that vacation must start at least g periods after the ending period of vacation in the previous vacation season. This can be modeled with equation

$$b_m \geq v_m + g, \qquad m = 1,2, \dots, M. \tag{3.35}$$

3.4.4. Objective function

The most important objective of the vacation planning is to maximize the minimum number of surplus workers w_{min}. This ensures that the surplus is split as evenly as possible in order to have room for unexpected events and absences such as sick leaves. Another objective is to minimize the maximum number of surplus workers w_{max}. The objectives can be presented in a single objective function

$$\max(w_{min} - qw_{max}), \tag{3.36}$$

where q is a parameter which can be altered to change the relative importance of the two optimization criteria.

4. Examples

In this chapter, the constraint programming formulation presented in the previous chapter is used to solve vacation planning problems. In chapter 4.1 a small problem with two workers is solved to illustrate how the constraint programming model works. After that, a description of two real world problems and results obtained with the constraint programming model are presented in chapters 4.2. and 4.3. The model was implemented for use in real world problems with IBM ILOG CPLEX CP Optimizer and solved using a computer with an Intel Core i7-4810Q CPU, 16GB of RAM memory and a 64-Bit operating system.

4.1. Step-by-step solution of a small problem

First the solution process of a small vacation planning problem is presented in detail. There are two drivers A and B, who both have a vacation allowance of 2 periods, and no special skills. Previous year's vacation starting period for driver A is 1 and for driver B it is 6. The vacation season in this example is 10 periods long, which include the end of one planning period (periods 1-4) and a full planning period (periods 5-10). There are no public holidays.

The constraints of the problem with the data of the instance are:

(C2) The length of the vacation of each driver m is 2 periods

(C3) Each worker's vacation must start at least $f = 2$ periods after the previous year's vacation start period.

(C5) The number of workers on vacation in each period is less than or equal to the maximum number p_n of workers allowed on vacation where $p = [3, 3, 4, 2, 3, 3, 2, 2, 2, 2]$.

(C9) Vacations cannot start in the second period of a planning period, i.e. in period $n = 6$.

(C13) The vacation of the first driver can shift forward at most $f_{max} = 4$ periods more than the minimum shift forward f (so the vacation must shift forward by 2 to 6 periods, see equation (3.23)).

The starting situation before allocating vacations is presented in Figure 7. An entry zero is used to mark periods with work for each worker and a yellow color indicates the position of previous year's vacation.

	1	2	3	4	5	6	7	8	9	10
A	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0
Max vacations	3	3	4	2	3	3	2	2	2	2
Vacations	0	0	0	0	0	0	0	0	0	0
Surplus	3	3	4	2	3	3	2	2	2	2

Figure 7: Base data

Before the vacations are assigned to the workers, the domain (i.e., possible values) for vacation variable's start period for each worker is computed. When constraints (C3) and (C13) are considered, $b_A \in [3, 4, 5, 6, 7]$. In addition, constraint (C9) forbids vacations to start in period 6, so the final domain for worker A's vacation start is $b_A \in [3, 4, 5, 7]$.

When constraints (C3) and (C9) are considered, the domain for the vacation variable's starting period for worker B is $b_B \in [8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20]$. Constraint (C9) forbids vacation start also in period 16, which corresponds to period 6 in the solution. Now, when the vacation starts in period 10 or 20, its ending point will be after the vacation season, which is not allowed (constraint (C11)). The domain of b_B is now reduced to $[8, 9, 11, 12, 13, 14, 15, 17, 18, 19]$. These are the possible starting periods for driver B's vacation when the vacation of driver A is not considered. But when the starting period of A's vacation is considered, constraint propagation can be used to reduce further the domain of b_B. Constraint (C4) states that the order of the vacations must stay the same as in the previous year. This is enforced with equations (3.21) and (3.22). Equation (3.21) states that the vacation of driver B cannot start before the vacation of driver A and equation (3.22) states that the vacation of driver B cannot go past the vacation of driver A after jumping. As the maximum for b_A is 7, the maximum for b_B is the corresponding period after jumping, period 17. Respectively, if $b_A = 5$, maximum for b_B is 15, if $b_A = 4$, maximum for b_B is 14 and if $b_A = 3$, maximum for b_B is 13. The decisions of assigning vacations can be formulated as a search tree with depth of three. The starting point of A's vacation is decided first and after that the starting point of B's vacation is decided:

- if $b_A = 3, b_B \in [8, 9, 11, 12, 13]$
- if $b_A = 4, b_B \in [8, 9, 11, 12, 13, 14]$
- if $b_A = 5, b_B \in [8, 9, 11, 12, 13, 14, 15]$
- if $b_A = 7, b_B \in [8, 9, 11, 12, 13, 14, 15, 17]$

All in all there are 26 possibilities for assigning the vacations. The results of all these vacation assignments are presented in Table 1.

Table 1: Results of all the feasible vacation assignments

b_A	b_B	w_{min}	w_{max}	$w_{min} - 0.1\, w_{max}$
3	8	1	3	0.7
3	9	1	3	0.7
3	11	1	3	0.7
3	12	1	3	0.7
3	13	0	3	-0.3
4	8	1	4	0.6
4	9	1	4	0.6
4	11	1	4	0.6
4	12	1	3	0.7
4	13	0	3	-0.3
4	14	0	4	-0.4
5	8	1	4	0.6
5	9	1	4	0.6
5	11	2	4	1.6
5	12	2	3	1.7
5	13	1	3	0.7
5	14	1	4	0.6
5	15	1	4	0.6
7	8	0	4	-0.4
7	9	1	4	0.6
7	11	1	4	0.6
7	12	1	3	0.7
7	13	1	3	0.7
7	14	1	4	0.6
7	15	1	4	0.6
7	17	0	4	-0.4

When the objective function is set to be max w_{min} (no weight is given to minimizing the maximum surplus) there are two optimal solutions: $b_A = 5, b_B = 11$ and $b_A = 5, b_B = 12$. The first solution is presented in Figure 8. Cells with a number 1 and green color represent vacations. Driver B's vacation jumps to the start of the vacation season. This is modeled by using a period number that is greater than 10. Figure 9 presents, where the vacation is really assigned, by considering each period $10 + \bar{n}$ as period n.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
Max vacations	3	3	4	2	3	3	2	2	2	2										
Vacations	1	1	0	0	1	1	0	0	0	0										
Surplus	2	2	4	2	2	2	2	2	2	2										

Figure 8: Optimal solution with objective function **max** $\boldsymbol{w_{min}}$

	1	2	3	4	5	6	7	8	9	10
A	0	0	0	0	1	1	0	0	0	0
B	1	1	0	0	0	0	0	0	0	0
Max vacations	3	3	4	2	3	3	2	2	2	2
Vacations	1	1	0	0	1	1	0	0	0	0
Surplus	2	2	4	2	2	2	2	2	2	2

Figure 9: The real assignment of vacations

If the objective function is changed to be max $w_{min} - 0.1w_{max}$, the second solution with $b_A = 5, b_B = 12$ is the only optimal solution. It has $w_{max} = 3$, which is one smaller than in the solution presented in Figures 8 and 9. This solution is presented in Figure 10.

	1	2	3	4	5	6	7	8	9	10
A	0	0	0	0	1	1	0	0	0	0
B	0	1	1	0	0	0	0	0	0	0
Max vacations	3	3	4	2	3	3	2	2	2	2
Vacations	0	1	1	0	1	1	0	0	0	0
Surplus	3	2	3	2	2	2	2	2	2	2

Figure 10: Optimal solution with objective function **max** $\boldsymbol{w_{min} - 0.1w_{max}}$

4.2. Real world problem 1

The first real world problem we consider consists in planning the vacations of one of the largest train driver bases in Finland. The plan is made for the fall vacation season. The personnel base has $M = 90$ drivers, and there are 2 special skills, 4 drivers have special skill 1 and 20 drivers have special skill 2. Table 2 presents the total vacation allowance (days) a_m of each driver m, the vacation allowance in periods $a_m/3$, previous year's summer vacation starting periods l_m and the possible special skills of each driver $s_{m,1}$ and $s_{m,2}$. The ending periods of the vacations in the previous vacation season are not considered.

Table 2: Base data about the drivers for fall vacation season

Driver	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
a_m	9	9	9	9	12	6	9	9	9	9	9	9	9	9	9	6	9	6	9	9	9	9	12	9	9	9	9	9	9	9
$a_m/3$	3	3	3	3	4	2	3	3	3	3	3	3	3	3	3	2	3	2	3	3	3	3	4	3	3	3	3	3	3	3
l_m	1	1	1	1	1	1	1	1	1	1	1	1	1	2	3	4	4	4	4	4	4	4	5	6	6	6	6	7	7	7
$s_{m,1}$	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$s_{m,2}$	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0

Driver	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
a_m	9	12	9	9	9	9	9	9	9	9	9	9	9	9	12	12	12	9	6	12	12	9	9	9	9	9	9	9	6	9
$a_m/3$	3	4	3	3	3	3	3	3	3	3	3	3	3	3	4	4	4	3	2	4	4	3	3	3	3	3	3	3	2	3
l_m	7	7	7	7	7	7	8	9	9	10	10	10	10	10	10	11	11	11	11	12	12	12	12	13	13	13	13	13	13	14
$s_{m,1}$	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$s_{m,2}$	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

Driver	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
a_m	12	12	12	9	9	9	9	9	9	9	6	6	12	12	12	9	9	9	9	9	6	6	9	9	9	9	9	9	9	9
$a_m/3$	4	4	4	3	3	3	3	3	3	3	2	2	4	4	4	3	3	3	3	3	2	2	3	3	3	3	3	3	3	3
l_m	18	18	18	18	18	18	18	18	18	18	18	18	20	21	21	21	21	21	21	22	22	22	22	22	22	22	23	23	23	23
$s_{m,1}$	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
$s_{m,2}$	0	0	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0

The fall vacation season includes 13 weeks, which correspond to $N = 26$ periods. The two of the first periods belong to planning period 15/2016, the next six periods (3-8) to the planning period 16/2016, and so on. There is one public holiday period ($n = 24,$ with $h(24) = 1$), which extends the overlapping vacations. Table 3 presents the maximum amount of workers allowed on vacation for each period p_n and the forbidden vacation starting periods with $h(n) = 1$.

Table 3: Base data about the periods for fall vacation season

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
p_n	22	22	21	19	19	19	19	19	19	19	19	19	19	19	15	15	15	15	15	15	20	20	20	20	20	20
$h(n)$	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0

Given the problem data above, the problem-specific constraints are now:

(C2) The length of the vacation of each driver m is a_m, unless the vacation overlaps a public holiday period. See Table 2 for the values of a_m.

(C3) Each worker's vacation must start at least $f = 8$ periods after the previous year's vacation start period in the fall vacation season.

(C5) The amount of workers on vacation in each period is less than or equal to the maximum amount of workers allowed on vacation p_n, see Table 3 for values of p_n.

(C8) The maximum amounts of workers with skills 1 and 2 allowed on vacation at the same time are $s_{max,1} = 1$ and $s_{max,2} = 5$.

(C9) Vacations cannot start in the second period of a planning period, i.e. in periods $n = 4, 10, 16, 22$

(C10) Vacations cannot start on public holiday period $n = 24$.

(C13) Vacation of the first driver can shift forward at most $f_{max} = 7$ periods more than the minimum shift forward f (so the vacation must shift forward 8 to 15 periods).

The base data was entered to the constraint programming model and the problem-specific constraints were altered to match the requirements presented in this chapter. The value $q = 0.1$ was used as the weight for minimizing the maximum surplus workers w_{max}, making maximizing minimum surplus workers w_{min} 9 times more important in the objective function.

An optimal solution was found in 31.99 seconds, having $w_{min} = 6$ and $w_{max} = 12$. A visualization of the solution is presented in Figures 13–14 in Appendix A. In the figures, rows represent drivers and columns represent periods. An entry 0 is used to mark working periods and 1 is used to mark vacation periods. The yellow color is used to mark previous year's vacations and the green color indicates the planned vacations. Drivers who have the special skill 1 are colored in blue and a bold font is used for drivers who have the special skill 2. Below the row of driver 90 there are rows that report different totals for each period: s is the amount of surplus workers, Vac is the amount of vacations planned, PVac is the amount of previous year's vacations, S1Vac is the amount of drivers with the special skill 1 on vacation and S2Vac is the amount of drivers with the special skill 2 on vacation.

The average surplus is 7.57, so the obtained minimum surplus of 6 is a good result considering the amount of constraints. The maximum surplus of 12 is quite high, but there are only 4 periods, where the surplus is higher than 10. No period has more than 1 worker with special skill 1 on vacation simultaneously and there is only 1 period which has 5 workers with special skill 2 on vacation simultaneously. The rest of the periods have 4 or less simultaneous vacations for drivers with skill 2. The vacations overlapping period 24 are one period longer than the vacation allowance, and no vacations start at period 24 or at the other forbidden starting periods. The vacations of drivers from 60 to 90 jump to the beginning of the vacation season.

The produced solution is sufficient for implementation, even though there are some "spikes" in the amount of reserve workers. These spikes could be used for example for planning periodical trainings. But as the model is fast in producing new solutions, it can be easily tested, which constraints are causing the spikes. After multiple test runs with different constraints relaxed, it was found out that maximum surplus could be reduced by removing the constraints that forbid vacation starting at periods 4 and 22. The reasons for this lowering of maximum surplus can be seen from Figure 14.

The last period ($n = 26$) has 12 surplus workers. If vacations of drivers 50 and 51 could be moved to start one period later at the forbidden staring period 22, the amount of surplus workers in period 26 would decrease to 10, while at the same time the amount of surplus workers in period 21 would increase from 7 to 9. In addition, period 4 has 12 surplus workers. Vacation of worker 77 cannot be shifted forward to start at period 3, because it would lower the minimum surplus, and it cannot start at period 4 because of the constraint. By allowing vacations to start in period 4, the surplus at period 4 can be lowered easily. The combination of allowing vacations to start at periods 4 and 22 allows also evening out the spikes at periods 8 and 10.

The results with the two relaxed constraints were obtained with calculation time of 56.19 seconds and they are presented in Appendix A, Figures 15 and 16. Now minimum surplus is $w_{min} = 6$ and maximum surplus is $w_{max} = 10$. Vacations starting from worker 57 jump to the beginning of the vacation season, 3 workers earlier than in the previous result. All the remaining constraints are respected; now 3 vacations start at period 4 and 7 vacations start at period 22. These can cause some difficulties at roster planning, so all in all the result with no relaxed constraints is better, even though w_{max} is higher by 2. In the test runs it was found out, that if the starting period of driver 1's vacation would not have an upper limit, a solution with $w_{min} = 7$ could be obtained, but the vacations shifted forward too much for practical applications. In the runs that were used to produce Figures 13-16, upper limit for the vacation start period of driver 1 was set to be 7 greater than the lower limit for the starting period.

4.3. Real world problem 2

The second real world problem we consider consists of planning vacations of a medium-sized personnel base for summer and fall vacation seasons. First, the vacation plan for the summer vacation season is made and after that the vacation plan of the fall vacation season is made while taking into account the summer vacation plan.

The personnel base has $M = 28$ drivers, and there are 9 drivers with a special skill. Table 4 presents the vacation allowance (days) a_m of each driver m, the vacation allowance in periods $a_m/3$, previous year's summer vacation starting periods l_m and the possible special skill of each driver $s_{m,1}$. The ending periods of the vacations in the previous vacation season are not considered.

Table 4: Base data about the drivers for summer vacation season

Driver	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
a_m	18	18	18	15	15	15	15	15	15	15	18	15	18	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
$a_m/3$	6	6	6	5	5	5	5	5	5	5	6	5	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
l_m	1	1	1	1	3	3	6	9	9	10	10	10	11	15	15	17	17	20	23	23	23	26	29	29	29	30	30	30
$s_{m,1}$	0	1	1	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	1	0	0	0	0

The summer vacation season includes 17 weeks, which consists of $N = 34$ periods. The first period belongs to the planning period 9/2016, the next six periods 2-7 belong to planning period 10/2016, the periods 8-13 to the planning period 11/2016 and so on. The last three periods 32-34 belong to planning period 15/2016. There is one public holiday period ($n = 7$), which extends the overlapping vacations. Table 5 presents the maximum amount of workers allowed on vacation for each period p_n and the forbidden vacation starting periods with $h(n) = 1$.

Table 5: Base data about the periods for summer vacation season

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
p_n	5	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7	7	7	7	7	5	5	5
$h(n)$	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0

Given the problem data above, the problem-specific constraints are now:

(C2) The length of the vacation of each driver m is a_m, unless the vacation overlaps a public holiday period. See Table 4 for the values of a_m.

(C3) Each worker's vacation must start at least $f = 12$ periods after the previous year's vacation start period in the summer vacation season.

(C5) The amount of workers on vacation in each period is less than or equal to the maximum amount of workers allowed on vacation p_n, see Table 5 for values of p_n.

(C8) The maximum amount of workers with skill 1 allowed on vacation at the same time is $s_{max,1} = 2$.

(C9) Vacations cannot start in the second period of a planning period, i.e., in periods $n = 3, 9, 15, 21, 27, 33$.

(C10) Vacations cannot start on the public holiday period $n = 7$.

The base data was entered to the constraint programming model and the problem-specific constraints were altered to match the requirements presented in this chapter. The value $q = 0.1$ was used as the weight of minimizing maximum surplus workers w_{max}, thus making the objective of maximizing the minimum surplus workers w_{min} 9 times more important than minimizing w_{max} in the objective function.

An optimal solution was found in 10.23 seconds, with having $w_{min} = 1$ and $w_{max} = 2$. A visualization of the solution is presented in Figure 11. In the figure, rows represent drivers and columns represent periods. An entry 0 is used to mark working periods and 1 is used to mark vacation periods. A yellow color is used to mark previous year's vacations and a green color the planned vacations. Drivers who have the special skill are colored in blue. Below the row of the driver 28 there are rows that report different totals for each period: s is the amount of surplus workers, Vac is the amount of vacations planned, PVac is the amount of previous year's vacations and SVac is the amount of drivers with the special skill on vacation.

The obtained solution is excellent, as the difference between minimum surplus workers and maximum surplus workers is only 1, while the constraint of having a maximum of two workers with the special skill on vacation simultaneously is also fulfilled. The vacations overlapping period 7 are one period longer than the vacation allowance. The vacations of drivers from 16 to 28 jump to the start of the vacation season, as putting more vacations at the end of the vacation season would lower w_{min}. No vacations start at any of the forbidden starting periods.

Driver		PP 10/2016						PP 11/2016						PP 12/2016						PP 13/2016						PP 14/2016						PP 15		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
16	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
s	2	2	2	1	1	1	2	1	2	2	1	1	1	1	1	2	2	1	1	1	1	2	2	2	2	2	2	2	1	2	2	1	2	2
Vac	3	4	4	5	5	5	4	5	4	4	5	5	5	5	5	4	4	5	5	6	6	5	5	5	5	5	5	5	6	5	5	4	3	3
Pvac	4	4	6	6	6	7	3	2	3	6	6	6	6	5	5	3	4	4	4	5	3	2	4	4	3	4	4	3	4	7	6	6	6	3
Svac	1	2	2	2	2	2	2	2	1	1	2	1	1	1	1	0	1	2	2	2	2	2	1	1	1	2	2	2	2	2	2	1	1	1

Figure 11: Results for the summer vacation season

Next, the vacation plan for the fall vacation season is made. One driver retires during the fall, so there are $M = 27$ drivers for which vacations are planned, 8 of which have the special skill. Table 6 presents the data for the fall vacation season. Now also the ending periods of summer vacations v_m are considered in making the vacation plan. The drivers are sorted by the vacation start period of previous year's vacations. As the order of previous year's vacations differs between summer and fall, the drivers are not in the same order as in the summer vacation season.

Table 6: Base data about the drivers for fall vacation season

Driver	1	3	5	6	2	12	9	11	4	10	20	21	7	18	23	24	25	26	28	19	17	16	22	8	14	15	27
a_m	12	12	9	6	6	9	9	9	6	6	6	9	9	9	9	9	9	9	9	6	9	6	9	6	6	6	6
$a_m/3$	4	4	3	2	2	3	3	3	2	2	2	3	3	3	3	3	3	3	3	2	3	2	3	2	2	2	2
l_m	37	37	39	39	41	41	41	43	45	45	47	47	49	49	51	51	51	53	53	55	55	55	56	58	58	58	58
$s_{m,1}$	0	1	0	0	1	0	1	1	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0
v_m	21	23	22	25	22	32	28	31	23	29	10	11	27	5	12	15	15	16	17	8	5	6	11	28	34	34	17

The fall vacation season includes 13 weeks, which consists of $N = 26$ periods. For the purpose of modeling the problem, the period numbers are set to start again from 1. In presenting this example, however, the period numbers are transformed so that they continue from the end of the summer vacation season. The first period is period number 36, as period 35 has two days in the summer vacation season and two days in the fall vacation season, so vacations cannot be planned in it. As no vacations are scheduled in period 35, it provides a good opportunity for trainings, as most of the drivers are at work. The first two periods belong to planning period 15/2016, the next six periods 38-43 to the planning period 16/2016, and so on. There is one public holiday period, which extends the overlapping vacations ($n = 59$). Table 7 presents maximum amount of workers allowed on vacation for each period p_n and the forbidden vacation starting periods with $h(n) = 1$.

Table 7: Base data about the periods for fall vacation season

n	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61
p_n	4	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6	5	5	5	5	5	5	5	5	5
$h(n)$	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0

Given the problem data above, the problem-specific constraints are now:

(C2) The length of the vacation of each driver m is a_m, unless the vacation overlaps a public holiday period. See Table 6 for the values of a_m.

(C3) Each worker's vacation must start at least $f = 8$ periods after the previous year's vacation start period in the fall vacation season.

(C5) The amount of workers on vacation in each period is less than or equal to the maximum amount of workers allowed on vacation p_n, see Table 7 for values of p_n.

(C8) The maximum amount of workers with skill 1 allowed on vacation at the same time is $s_{max,1} = 2$.

(C9) Vacations cannot start in the second period of a planning period, i.e. in periods $n = 39, 45, 51, 57$.

(C10) Vacations cannot start on a public holiday period $n = 59$.

(C12) Vacation of each driver m must start at least $g = 10$ periods (5 weeks) after the ending period summer vacation v_m.

The base data was inputted to the constraint programming model and the value $q = 0.1$ was used as the weight for minimizing the maximum surplus workers w_{max}. An optimal solution for the problem was found in 7.50 seconds, having $w_{min} = 2$ and $w_{max} = 3$. A visualization of the solution is presented in Figure 12. In this figure, next to the driver-column is also reported a column gap which indicates the gap from the end of the summer vacation to the start of the fall vacation in periods.

The solution is again very good; the difference between the minimum surplus workers and the maximum surplus workers is only 1, and there is a maximum of two workers with the special skill on vacation simultaneously. The vacations overlapping period 59 are extended to be one period longer than the vacation allowance. There are not vacations that start at the forbidden starting periods. The smallest gap between the end of summer vacation and the start of fall vacation is 11 periods, which is acceptable.

Driver	Gap	PP 15		PP 16/2016						PP 17/2016						PP 1/2017						PP 2/2017					
		36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61
1	24	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
3	24	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	25	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
6	22	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
2	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
12	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
9	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
11	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
4	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
10	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
20	45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
21	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
7	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
18	52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
23	23	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	21	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	21	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	23	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	22	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	31	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	37	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	36	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	31	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	15	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	11	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	11	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	28	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	s	3	2	2	3	2	2	3	2	2	3	2	2	2	2	2	3	3	3	2	3	2	3	2	3	3	3
	Vac	1	3	3	2	3	3	2	3	4	3	4	4	4	4	4	3	3	2	3	2	3	2	3	2	2	2
	Pvac	0	2	2	4	4	4	3	3	1	3	2	2	2	3	2	5	3	5	2	5	4	2	5	4	0	0
	Svac	0	1	1	1	1	1	0	1	1	1	0	0	1	1	2	2	2	2	2	0	1	1	1	0	0	0

Figure 12: Results for the fall vacation season

5. Experiences from scheduling real world vacations

The vacation planning model presented in Chapter 3 was used in planning the vacations of the train drivers for the year 2016 at VR. Plans for the spring vacation season were made in September 2015, and plans for the summer and fall vacation season were made in February 2016. This chapter summarizes the experiences of planning vacations with the new model.

The first step of the vacation planning is to gather the base data, which was not a simple task. Forecasting demand for freight trains many months in advance is especially difficult. A spreadsheet was sent to the superior of each personnel base in order to gather information about the retirements and fixed absences such as parental leaves. At the same time, the preferences regarding vacation allocations of drivers with short vacation allowance were asked, as they had two possibilities for allocating their vacations between the vacation seasons. After that, the resource planners filled in the tables reporting the special skills of the workers. The actual start of the vacation planning was then delayed as much as possible to obtain better demand forecasts.

After the base data was collected, it had to be preprocessed in order to be given in input to the constraint programming model. This included for example converting previous year's (2015) vacation start dates into vacation start periods corresponding to periods of the 2016 vacation seasons. Information about the vacations of retiring drivers and fixed absences of other drivers was also expressed in terms of periods and it was combined with the demand forecast to obtain the maximum allowed vacations for each period. Finally, the drivers were ordered according to their previous year's vacation start period to allow a simple formulation of constraint (C4).

The constraint programming model was used in planning the vacations of all the personnel bases. For most of the cases, good solutions were found in less than a minute, while the solution time for the largest personnel bases was at most 3 minutes. The largest personnel base had 124 drivers and the smallest personnel base had 3 drivers. The spring vacation season had 40 periods in which vacations were planned, the summer vacation season had 34 periods, and the fall vacation season had 26 periods. In some of the personnel bases an applicable solution was found with the first run of the model, but quite many of the personnel bases needed some tweaking of the constraints or the base data in order to improve the solution quality. Often, in the largest personnel bases, the maximum surplus workers tended to be much larger than the minimum surplus. The last period of a vacation season had usually less vacations than the previous periods, as vacations have different lengths, but a vacation of a driver cannot start before the vacation of the previous driver. The forbidden vacation starting periods posed also challenges occasionally, causing “spikes” in the surplus drivers for some periods, but usually those did not hamper the solution quality much. The gap constraint (C12) also caused difficulties in some of the personnel bases. The gap from the previous vacation had to be at least 4 weeks (8 periods), and vacations had to shift forward 7 weeks in the spring vacation season, 6 weeks in the summer vacation season and 4 weeks in the fall vacation season.

As an example, a driver could have had his/her spring vacation ending at the end of May, and the previous year’s summer vacation starting at the beginning of August. As a summer vacation must shift forward by at least 6 weeks with respect to the previous year, the earliest possible starting time for the vacation would be at the middle of September. But as a summer vacation is usually 3 weeks long, the vacation would not fit entirely within the summer vacation season (June-September) even if starting at the earliest possible period, so it must jump to the beginning of the summer vacation season. Now the constraint (C12) forces the vacation to start at least 4 weeks after the start of the summer vacation season, so the earliest possible starting time for the vacation would be at the beginning of July. Overall, the constraints force the vacation to shift forward by three months, while the summer vacation season is only four months long. This is not a desirable result. In addition, the large shift forward of this vacation forces all the vacations that must start after it to be also shifted forward significantly. In some cases, situations like this one required to relax the constraint (C4) regulating the order of the vacations start times, while the constraint (C12) was still always imposed.

After the vacation plans were ready, drivers were given two weeks of time to review them and to make requests for changes. Depending on the personnel base, 0 to 50 % of the drivers requested changes for their vacations. The largest percentages were on a few of the smallest personnel bases. On average around 10-20 % of drivers made requests. The requests for changing vacation timings had to be accepted or denied by manually adjusting the optimized vacation plans. This task consumed quite a lot of time, but allowing the drivers to request changes was important for them, as the new vacation planning model reduced the predictability of their vacation timings.

As the base data had to be processed manually to be in the form required by the model, a few mistakes did happen. There were a few personnel bases where the vacation allowances of the drivers got mixed, and this was not noticed until the preliminary vacation plans were released to the drivers. As the vacation times were already released, a complete new plan with correct data could not be run with the model. The vacation lengths had to be corrected by hand while keeping the starting periods unchanged.

All in all the vacation planning process required much less time after the introduction of the new planning model. Gathering and processing the base data was the most time-consuming phase of the planning, whereas the plans could be produced quickly with the model. The small running times allowed testing multiple combinations of constraints in order to get as good as possible results. For example, if a good result was obtained while imposing that a maximum of 5 workers with a certain special skill can be on vacation simultaneously, a new run could also be made while imposing a smaller maximum of 4 drivers. This allowed checking whether the other characteristics of the solution stay satisfactory while the availability of workers with that special skill is increased.

At the time of finalizing this thesis, the vacations of train drivers for year 2017 were being planned. Solution times to optimal solution for the three vacation seasons were recorded for all the personnel bases, and those are presented in Table 8. The table includes personnel base number, number of drivers working at the personnel base, number of different special skills at the personnel base, and solution time (ST) in seconds for the spring-, summer-, and fall vacation season. The number of drivers varies between 116 and 3 while the number of special skills varies from 10 to 0. The winter vacation season has 43 periods, the summer vacation season 35 periods and the fall vacation season 26 periods.

All the solution times were under 50 seconds, making the solution times negligible compared to the time needed for gathering base data. The solution times decrease when the number of drivers in the personnel base decreases. Between the vacation seasons the solution times are similar, even though the spring vacation season has 17 more periods than the fall vacation season. The number of special skills does not impact solution times much either.

Table 8: Solution times for all the personnel bases

Personnel base	Number of drivers	Special skills	ST spring (seconds)	ST summer (seconds)	ST fall (seconds)
1	116	10	34.50	39.30	45.99
2	99	7	27.66	33.27	23.83
3	90	2	27.46	28.39	19.76
4	61	1	15.81	17.41	17.88
5	45	3	11.48	10.03	8.84
6	44	3	9.45	8.99	15.50
7	44	2	7.76	8.38	8.22
8	42	3	17.07	13.17	11.33
9	39	3	11.28	8.55	12.62
10	38	4	5.27	8.34	6.27
11	35	1	10.71	9.90	8.30
12	29	4	7.23	5.15	4.65
13	28	2	11.64	15.18	13.09
14	27	2	13.19	7.93	7.27
15	24	2	5.50	5.24	6.40
16	16	2	7.94	7.19	6.95
17	13	1	4.76	5.39	7.39
18	10	1	7.92	8.57	7.70
19	10	1	8.05	7.21	8.66
20	9	1	4.26	5.01	5.35
21	6	1	4.55	4.58	4.48
22	3	0	4.69	4.01	4.38

6. Conclusions and discussion

The objective of this thesis was to develop an automated solution approach for planning train drivers' vacations, reduce the time needed for vacation planning, and reduce the overtime costs of the company. A constraint programming formulation proved to be an efficient way of modeling the problem. It allowed including various constraints that were used to improve the solution quality for all the stakeholders. The employees for whom the vacations are planned have different preferences with regard to the vacation planning and the same is true for the planning personnel or the management of the company.

Drivers' union representatives highlighted that the new vacation planning process must be fair for all the drivers, and that the vacations must circulate among the drivers as was the case in the previous model. The circulation goal was achieved to some extent by introducing a constraint imposing a minimum shift forward for each vacation start with respect to the previous year. Two contributing factors to the fairness of the plans were keeping the order of the vacations the same as in the previous year, and allowing each driver an equal chance to ask for a change in his/her vacation dates after the preliminary plans were released.

The planning personnel needed a tool that would alleviate the workload related to the vacation planning. As the plans were previously made on spreadsheets by hand, the process was time-consuming and prone to errors. Also, making changes to the plan after spotting an error was a lengthy process as a change in a single driver's vacation often caused a need to alter multiple other vacations in order to balance the reserve drivers over all periods. The new model reduced the planning time needed considerably, and decreased the amount of errors. As new vacation plans could be made in minutes, the developed model also allowed testing multiple scenarios with slightly different values for the constraints in order to search the best possible solution with respect to different characteristics of the solution.

The company wanted to lower overtime costs, and optimizing the allocation of drivers' vacations was found to be an efficient solution. With the previous vacation planning method, vacation levels did not vary inside a vacation season, whereby too many vacations were scheduled in planning periods with low maximum working time (for example around Christmas), thus causing large overtime costs. The new vacation planning model allowed taking into account work demand forecasts at planning period level, and this enabled assigning a much lower amount of vacations in the planning periods with public holidays. In theory this reduces overtime costs caused by vacation allocation drastically, but there is not yet enough data about the effects of adopting the new model to allow drawing conclusions about this.

Drivers' union representatives prefer the old vacation scheduling process, which was simpler to understand and deemed to be fairer. The old process was more constrained than the new approach. Drivers were placed on vacation in groups of constant size. All the drivers in a vacation group had vacation at the same time, and the timings of the vacations of the following year were also precisely known in advance. The work amount forecasts used in the new approach are not very accurate, as those must be made months before the duties and rosters are planned for drivers. Some of the aspects of the actual demand and training needs are also not known at the time of vacation planning. For example, freight traffic volumes depend on customer demand and even on weather. As the forecasts cannot be as accurate as needed, some of the work must be transferred between depots during the actual planning of the duties for the drivers which takes place just a few weeks before drivers actually perform the duties.

Still, the introduction of the new vacation planning process has been a success; there has been surprisingly little resistance from the drivers, although the new process lowered the predictability of vacation timings. Clear visualizations of the solutions have been really important. Those allow the drivers to see that they are treated equally, as the order of the vacations stays the same as in the previous year and the vacations shift forward by approximately the same amount of weeks from the previous year's vacations. The planning personnel also prefer the new model, although gathering and processing the base data for the optimization model is still a time-consuming task. Also, dealing with the requests of change in vacation timings takes some time. The management of the company appreciates the new vacation planning process, as it allows more freedom in vacation planning. All in all the implementation of the new vacation planning model and process went well, and the goals set in the beginning of the project were achieved.

One of the possible future developments for the model is to allow planning for the whole year (three vacation seasons) simultaneously. This could lead to better solutions at a yearly level as vacations are not planned in three separate seasons. The model could be also adapted to plan the vacations of different employee groups, for example conductors. This should be relatively simple, as constraint programming allows modifying the constraints easily. Lastly, the constraint (C13) reducing the maximum shift forward of first driver's vacation could be formulated as a soft constraint, adding a related penalty term to the objective function. The constraint could be imposed for all the drivers, and both linear and nonlinear penalty functions for vacations shifting forward too much could be tested to improve solution quality.

References

[1] Al-Zubaidi, H. & Christer, A.H. 1997 "Maintenance manpower modelling for a hospital building complex", *European Journal of Operational Research*, vol. 99, no. 3, pp. 603-618.

[2] Azmat, C.S., Hürlimann, T. & Widmer, M. 2004 "Mixed Integer Programming to Schedule a Single-Shift Workforce under Annualized Hours", *Annals of Operations Research*, vol. 128, no. 1-4, pp. 199-215.

[3] Azmat, C.S. & Widmer, M. 2004 "A case study of single shift planning and scheduling under annualized hours: A simple three-step approach", *European Journal of Operational Research*, vol. 153, no. 1, pp. 148-175.

[4] Baker, K.R. & Magazine, M.J. 1977, "Workforce Scheduling with Cyclic Demands and Day-Off Constraints", *Management Science*, vol. 24, no. 2, pp. 161-167.

[5] Bartholdi, J., Orlin, J. & Ratliff, H. 1980, "Cyclic Scheduling via Integer Programs with Circular Ones." *Operations Research*, vol. 28, no. 5, pp. 1074-1085.

[6] Beaulieu, H., Ferland, J., Gendron, B. & Michelon, P. 2000, "A mathematical programming approach for scheduling physicians in the emergency room.", *Health Care Management Science*, vol.3, no. 3, pp. 193-200.

[7] Beaumont, N. 1997, "Using Mixed Integer Programming to Design Employee rosters" *The Journal of the Operational Research Society*, vol. 48, no. 6, pp. 585–590.

[8] Bellanti, F., Carello, G., Della Croce, F. & Tadei, R. 2004 "A greedy-based neighborhood search approach to a nurse rostering problem", *European Journal of Operational Research*, vol. 153, no. 1, pp. 28-40.

[9] Billionnet, A. 1999 "Integer programming to schedule a hierarchical workforce with variable demands", *European Journal of Operational Research,* vol. 114, no. 1, pp. 105-114

[10] Burns, R. 1978, "Manpower Scheduling with Variable Demand and Alternate Weekends off", *INFOR*, vol. 16. no. 2, pp. 101-111.

[11] Carter, M.W. & Lapierre, S.D. 2001, "Scheduling emergency room physicians" *Health care management science,* vol. 4, no. 4, pp. 347-60.

[12] Cheng, B.M.W., Lee, J.H.M. & Wu J.C.K. 1996 "A Constraint-Based Nurse Rostering System Using a Redundant Modeling Approach", *Eighth IEEE International Conference on Tools with Artificial Intelligence*, pp. 140–148.

[13] Cheng, B.M.W., Lee, J.H.M. & Wu J.C.K. 1997 "A Nurse Rostering System Using Constraint Programming and Redundant Modeling", *IEEE Transactions on Information Technology in Biomedicine*, vol. 1, no. 1, pp. 44-54.

[14] Chong, P.S. & Strevell, M.W. 1985, "A vacation scheduling algorithm for military flight crews: Maximizing satisfaction while maintaining military preparedness", *Journal of Operations Management*, vol. 5, no. 2, pp. 205-211.

[15] Corominas, A., Lusa, A. & Pastor, R. 2007 "Using a MILP model to establish a framework for an annualised hours agreement", *European Journal Of Operational Research*, vol. 177, no. 3, pp. 1495-1506.

[16] Costa, M.-C., Jarray, F. & Picouleau, C. 2006 "An acyclic days-off scheduling problem", *4OR*, vol. 4, no. 1, pp. 73-85.

[17] Day, P.R. & Ryan, D.M. 1997 "Flight Attendant Rostering for Short-Haul Airline Operations", *Operations Research*, vol. 45, no. 5, pp. 649-661.

[18] Dewess, S. 2010 “Socially acceptable annual holiday planning for the crew of a local public transport company in Germany”, *Public Transport*, vol. 2, no. 1, pp. 25-49.

[19] Emmons, H. & Burns, R.N. 1991 “Off-Day Scheduling with Hierarchical Worker Categories”, *Operations Research*, vol. 39, no. 3, pp. 484-495.

[20] Emmons, H. & Fuh, D.-S. 1997 “Sizing and Scheduling a Full-Time and Part-Time Workforce with off-day and off-weekend constraints”, *Annals of Operations Research*, vol. 70, pp. 473-492.

[21] Ernst, A.T., Jiang, H., Krishnamoorthy, M., Owens, B. & Sier, D. 2004a “An Annotated Bibliography of Personnel Scheduling and Rostering”, *Annals of Operations Research*, vol. 127, no. 1, pp. 21-144.

[22] Ernst, A.T., Jiang, H., Krishnamoorthy, M. & Sier, D. 2004b “Staff scheduling and rostering: A review of applications, methods and models”, *European Journal of Operational Research*, vol. 153, no. 1, pp. 3-27.

[23] Fahle, T., Junker, U., Karisch, S.E., Kohl, N., Sellmann, M. & Vaaben, B. 2002, “Constraint programming based column generation for crew assignment”, *Journal of Heuristics*, vol. 8, no. 1, pp. 59-81.

[24] Gärtner, J., Wahl, S. & Hörwein, K. 1998 “A Technique to Take Leave into Account in Shift-Rota design”, *Scandinavian Journal Of Work Environment & Health*, vol. 24 no. 3, pp.103-108.

[25] He, F. & Qu, R. 2012, “A constraint programming based column generation approach to nurse rostering problems”, *Computers & Operations Research*, vol. 39, no. 12, pp. 3331-3343.

[26] Hertz, A., Lahrichi, N. & Widmer, M. 2010 “A flexible MILP model for multiple-shift workforce planning under annualized hours”, *European Journal Of Operational Research*, vol. 200, no. 3, pp. 860-873.

[27] Hung, R. 1994 “Single-shift off-day scheduling of a hierarchical workforce with variable demands”, *European Journal of Operational Research*, vol. 78, no. 1, pp. 49-57.

[28] Hung, R. 1999 “Scheduling a workforce under annualized hours”, *International Journal Of Production Research*, vol. 37, no. 11, pp. 2419-2427.

[29] Knust, S. & Schumacher, E. 2011, “Shift scheduling for tank trucks”, *Omega -international Journal Of Management Science*, vol. 39, no. 5, pp. 513-521.

[30] Koutsopoulos, H.N. & Wilson, N.H.M. 1987 “Operator workforce planning in the transit industry”, Transportation Research Part A: General, vol. 21, no. 2, pp. 127–138.

[31] Laborie, P. & Rogerie, J. 2008 “Reasoning with Conditional Time-intervals” *Proceedings of the 21th International Florida Artificial Intelligence Research Society Conference, FLAIRS-21*, pp. 555–560.

[32] Laborie, P., Rogerie, J., Shaw, P. & Vilím, P. 2009 “Reasoning with Conditional Time-intervals, Part II: an Algebraical Model for Resources.” *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference, FLAIRS-22*, pp. 201-206.

[33] Morris, J. & Showalter, M. 1983, “Simple Approaches to Shift, Days-off and Tour Scheduling Problems.” *Management Science,* vol. 29, no.8, pp. 942–950.

[34] Narasimhan, R. 1997 “An Algorithm for Single Shift Scheduling of Hierarchical Workforce”, *European Journal Of Operational Research*, vol. 96, no. 1, pp.113-121

[35] Nurmi, K., Kyngäs, J. & Post, G. 2011 “Driver rostering for bus transit companies”, *Engineering Letters*, vol. 19, no. 2, pp. 125-132.

[36] Sellmann, M., Zervoudakis, K., Stamatopoulos, P. & Fahle, T. 2002, “Crew assignment via constraint programming: Integrating column generation and heuristic tree search”, *Annals of Operations Research*, vol. 115, no. 1-4, pp. 207-225.

[37] Thompson, G.M. 1998a "Labor scheduling, part 1: Forecasting demand", *Cornell Hotel and Restaurant Administration Quarterly*, vol. 39, no. 5, pp. 22-31.

[38] Thompson, G.M. 1998b "Labor scheduling, part 2: Knowing how many on-duty employees to schedule", *Cornell Hotel and Restaurant Administration Quarterly*, vol. 39, no. 6, pp. 26-37.

[39] Thompson, G.M. 1998c "Labor scheduling, part 3: Developing a Workforce schedule", *Cornell Hotel and Restaurant Administration Quarterly*, vol. 40, no. 1, pp. 86-96.

[40] Thompson, G.M. 1998d "Labor scheduling, part 4: Controlling workforce schedules in real time", *Cornell Hotel and Restaurant Administration Quarterly*, vol. 40, no. 3, pp. 85-96.

[41] Van Den Bergh, J., Belieen, J., De Bruecker, P., Demeulemeester, E. & De Boeck, L. 2013 "Personnel scheduling: A literature review" *European Journal of Operational Research*, vol. 226, no. 3, pp. 367-385

[42] Walsh, T. 1997 "Depth-bounded discrepancy search" *Proceedings of IJCAI-97*, pp. 1388–1393.

[43] Yunes, T.H., Moura, A.V. & de Souza, C.C. 2005, "Hybrid column generation approaches for urban transit crew management problems", *Transportation Science*, vol. 39, no. 2, pp. 273-288.

Appendix A: Results of real world problem 1

Driver	PP 15		PP 16/2016						PP 17/2016						PP 1/2017						PP 2/2017					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0

Figure 13: Results of real world problem 1, part 1/2

43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
60	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
61	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
63	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
64	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
66	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
67	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
68	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
69	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
71	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
73	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
75	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
76	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
77	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
79	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
81	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
82	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
83	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
86	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
87	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
88	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
89	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
s	9	8	6	12	8	8	7	12	6	11	6	6	6	8	6	7	6	6	6	6	7	7	6	8	7	12
Vac	13	14	15	7	11	11	12	7	13	8	13	13	13	11	9	8	9	9	9	9	13	13	14	12	13	8
Pvac	13	14	14	10	9	10	14	15	12	10	12	14	14	13	8	1	0	12	12	11	10	14	18	11	4	0
S1vac	1	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0
S2vac	**4**	**4**	**5**	**1**	**1**	**2**	**3**	**3**	**5**	**3**	**3**	**3**	**3**	**3**	**2**	**2**	**1**	**0**	**0**	**0**	**2**	**2**	**3**	**2**	**3**	**2**

Figure 14: Results of real world problem 1, part 2/2

Driver	PP 15		PP 16/2016						PP 17/2016						PP 1/2017						PP 2/2017					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Figure 15: Results with relaxed constraints, part 1/2

47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
57	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
58	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
59	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
61	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
63	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
64	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
66	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
67	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
68	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
69	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
71	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
73	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
75	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
76	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
77	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
79	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
81	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
82	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
83	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
86	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
87	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
88	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
89	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
s	10	8	7	10	10	7	7	10	6	6	6	6	7	8	6	6	7	9	6	6	10	6	6	9	9	10
Vac	12	14	14	9	9	12	12	9	13	13	13	13	12	11	9	9	8	6	9	9	10	14	14	11	11	10
Pvac	13	14	14	10	9	10	14	15	12	10	12	14	14	13	8	1	0	12	12	11	10	14	18	11	4	0
S1vac	1	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0
S2vac	4	5	4	2	1	2	2	3	5	5	3	2	2	3	2	2	1	1	1	0	1	2	3	2	2	2

Figure 16: Results with relaxed constraints, part 2/2