USING MATHEMATICAL MODELLING TO CREATE A DRIVER SCHEDULE WHILE TAKING SCHEDULING RESTRICTIONS INTO ACCOUNT

MSc Thesis Industrial Engineering & Management

Supervisors University of Twente:
DR.IR. J.M.J. SCHUTTEN
DR.IR. E.A. LALLA
23-09-2021

Faculty: Behavioural, Management and Social Sciences
Study: Industrial Engineering and Management
Track: Production and Logistics Management
Research Orientation: Supply Chain and Transportation Management

UNIVERSITY OF TWENTE.

Morrenhof, Max
S2205793

Management summary

Introduction

We conduct this study at a distribution center of an anonymous company operating in the food industry. The implementation of a new advanced route planning system forces the company to reorganize its process of scheduling drivers. Since the process of scheduling drivers should be reorganized, right now is the perfect moment to optimize this process as well. So, the company wants to be ready for the future and create a method that schedules their drivers optimally and that can be used along with the new advanced planning system.

Problem statement

We use a problem cluster to identify the core problem from the central problem. The central problem is that there is no possibility to schedule personnel optimally. The current method of scheduling drivers is a time-consuming and failure sensitive task, because there is no standard method and a lot of manual actions are needed to create a schedule.

We formulate the following main research question in this research:

> ***"What method should the company use to create a weekly schedule for its drivers that can be used along with the advanced planning system with the aim of lowest cost possible?"***

Approach

First, we analyse the current situation by the use of several data collection methods such as interviews and data analysis. The transport department performs a lot of manual actions to create their driver schedule. They start with creating a block schedule, which takes approximately 8 hours. The block schedule is a visual representation of the route plan. The block schedule is created to have a clear overview of all properties of the shifts that have to be executed. From there, the transport planners create a driver schedule per week. While creating a driver schedule, the operational transport planners have to take into account several restrictions of the drivers. These restrictions are: contractual hours, start time, end time, total duration per shift, skills and workload. Creating a complete weekly driver schedule takes 40 hours. So, in total it costs 48 hours to create a driver schedule from scratch.

The schedule has an average deviation of 2 hours and 26 minutes between scheduled and contractual hours per driver. This is undesirable since hours lower than the contractual hours are paid, but not worked. Hours above the contractual hours are overtime hours that have to be paid out with an overtime percentage.

The literature contains multiple problems that have common grounds with our problem. We describe general problems and aspects from scheduling and rostering. The nurse scheduling problem, the airline crew scheduling problem and the bus driver problem have overlap with our problem. Modelling features that overlap are preferences of employees, the conflicting interest between the employees and the organization and that obtaining good solutions quickly is important. The literature study about optimization methods shows that both heuristics and mathematical modelling are frequently used techniques. Heuristics are mainly used when problems are not solvable using exact techniques (NP-hard).

Our problem has common grounds with a set partitioning problem, where one employee per duty needs to be scheduled. However, in our case, not all shifts need to be filled, which is a situation not described in the literature. We formulate a mathematical program with the objective of minimizing the deviation between scheduled and contractual hours. We formulate all restrictions of the drivers as constraints. We use pre-processing techniques to reduce the problem size and speed up the running times.

Results

We use 3 different models to simulate several options. In the first model we consider all restriction as hard constraints. This means that there is no option to violate the given restrictions. We allow paid waiting time in the second model. When we allow paid waiting time, it means that the option exists that a driver starts after his or her maximum start time. However, the driver will get paid the time he or she is waiting. In the last model we consider the restrictions regarding the start times to be soft constraints. This is done since the restrictions regarding start time are rather preferences than hard constraints and thus it is possible to violate them.

The disadvantage of the model with hard constraints is that it cannot always provide a feasible solution, since there does not always exist a solution for all instances. We observe that the model with soft constraints regarding the start times performs the best with an average reduction of 95.8% in deviation between scheduled and contractual hours compared to the current situation. The analysis of 2020 confirms what we conclude over the other 11 scenarios. Also in 2020, the model with soft constraints performs the best, with an average reduction of 95.7% in deviation between scheduled and contractual hours compared to the current situation. The reduction in the model with soft constraint comes with the cost that some restrictions regarding start times are violated.

We compare a commercial license-based solver (IBM CPLEX) with a free solver (Python MIP). The license-based solver outperforms the free solver significantly. The license-based solver finds in 5 minutes better solutions than the free solver does in 1 hour.

We perform a Monte Carlo simulation using real data to simulate the execution of a schedule. The purpose of the simulation is to see what the deviation between realized and contractual hours is after the execution of the schedule. The simulation shows that each scenario results in a total deviation of between 130 and 180 hours. This worse result is due to the fact that the real data shows that it takes on average 7% more time to execute a shift than scheduled.

The analysis of the schedules shows that 2 specific drivers have a relatively big negative influence on the solution values. These drivers are drivers with a contract containing 46 working hours in 4 days. This means that they should have an average of 11.5 working hours per day. These drivers influence the deviation between scheduled and contractual hours on average 59.2% per week.

Recommendations

We recommend the company to start using the model as a tool to help the planner create weekly schedules. Since the model with soft constraints performs the best in our experiments, we recommend using this model. We emphasize that both the model and the planner should be used in their strengths, the model for its computational power and the planner for its human intuition to deal with uncommon situations. Since the schedule is on a tactical level, we use a running time of 1 hour. Using the model saves 48 labour hours per schedule and realizes a reduction of 95.8% in deviation between scheduled and contractual hours compared to the current situation.

Next, we recommend taking actions on the drivers with 46 contractual hours. 2 drivers with a contract of 46 hours have a relatively big negative influence on the deviation between scheduled and contractual hours: 59.2%. Our recommendation is to offer these drivers a 40-hour contract or let them work 5 days instead of 4. Another option can be to let the Supply Chain Planner provide longer shifts, especially for these drivers.

The last recommendation is a recommendation regarding the schedule robustness. Our sensitivity analysis shows that the realized hours are on average 7% above the scheduled hours. To improve the schedule robustness, we recommend either to schedule fewer hours than the contractual hours or to better estimate the duration of the shifts.

Preface

Dear reader,

With great pleasure I present you the result of my research conducted at an anonymous company operating in the food industry. This thesis has been written to finish the master Industrial Engineering & Management, with the specialization Production & Logistics Management. I want to express my gratitude to some people who helped me during my thesis and my studies.

First, I would like to thank my supervisor(s) from the company. I cannot mention their names, since these can be related to the company, which is anonymous. However, they provided me valuable feedback and brought me in contact with valuable people for my project. I also want to thank my colleagues from production planning, I always could have a laugh with them. Finally, I want to thank the transport department for their input for this research.

Moreover, I would like to thank both my first and my second supervisor from the university. Marco Schutten was my first supervisor, he helped me through the process from the begin and provided valuable feedback on my thesis. Eduardo Lalla-Ruiz joined later in the process and helped me with his expertise about mathematical modelling.

Lastly, I want to thank my family, friends and my fellow students from the university who supported my during my thesis and the study. I could always count on them to have a laugh, discussion or talk about concerns.

Enjoy reading the report!

Max Morrenhof

Beilen, September 23, 2021

Table of Contents

List of figures

List of tables

1. Introduction

This report describes the result from the Master IEM graduation at the University of Twente at an anonymous company. The research focuses on optimizing the driver schedule. The need for optimization of the driver schedule is a result of the introduction of a new advanced planning system.

This chapter starts with a company introduction and a description of the departments where the problem exists in Section 1.1. Section 1.2 gives an overview regarding planning and scheduling within the company. Sections 1.3 and 1.4 describe the research motivation and problem statement, respectively. Furthermore, Section 1.5 describe the objective of the research. The chapter finalizes with Section 1.6 describing the research design.

1.1 Company introduction

This section gives an introduction to the company. First, Section 1.1.1 introduces the company itself and gives a short description of the process. Section 1.1.2 describes the departments involved in the research.

1.1.1. Introduction

The research is conducted in a fast-growing company in the food industry. The company has around 700 stores in the Netherlands and a few distribution centers to supply these stores. This research is carried out at one of these distribution centers. At this distribution center, the company receives products from suppliers and redistributes these to their stores. The distribution center picks two types of products: fresh/cold products and non-fresh products. Furthermore, the distribution center also functions as a 'cross-dock'. Cross-docking means that containers with products coming in, are already sorted on store level. Figure 1 shows a short description of the main process:

Figure 1: Company process

First the stores place orders. These orders are based on automatic replenishment of products and operational adjustments. These orders are visible for the planning department after the cut-off moment. The cut-off moment is the last moment at which stores can place orders for specific time windows. When the orders are definitive, they become visible in the system for the planners. A first version of the routes is already planned based on a forecast. The planners replan the routes when the definitive orders are in the system. Then the planners release the orders for production. Releasing for production means that the orders are ready to be picked. Then the orders are picked on containers and these containers are placed on the dock. Finally, after completion of all orders, the truck is loaded at the distribution center and later unloaded at the store.

1.1.2. Departments

Figure 2 shows a simplified version of the organogram of the company. The research has common ground with two departments. These departments are the transport department and Production Planning. Production Planning is a sub-department of Site Support. Site Support is focused on supporting the warehouse and its processes. Site Support consists mainly of office jobs and is responsible for aspects such as planning, warehousing and IT.

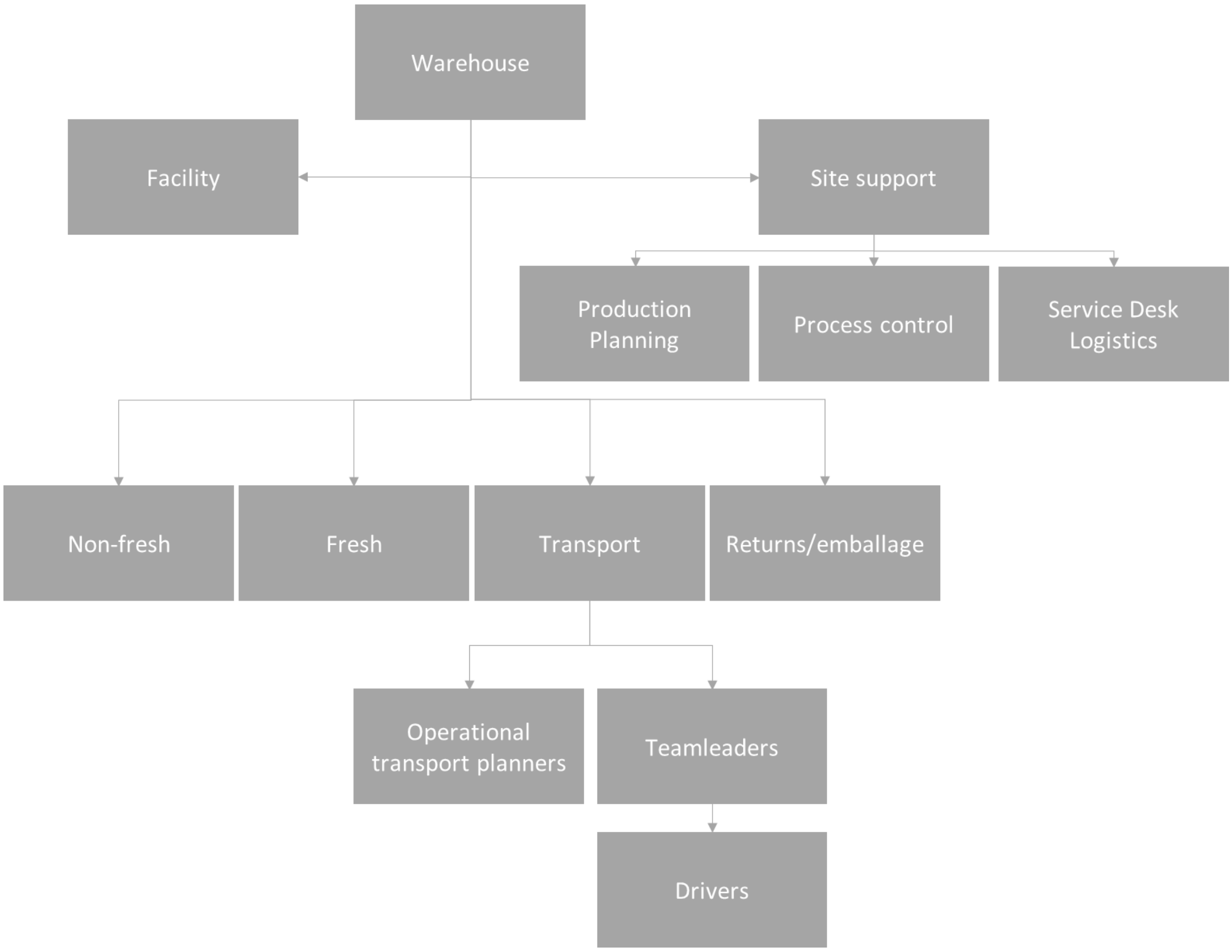

Figure 2: Organogram

The transport department is responsible for the outgoing transport. So, the orders are picked at the distribution center and are collected on the dock. From the dock, the containers are loaded into the truck after the driver has arrived. From that point, the transport department is responsible for the remainder of the process.

The transport department consists of 2 team leaders, 8 operational transport planners and around 90 drivers. For the outgoing transport, the company also owns a truck fleet of around 50 trucks. However, these drivers and trucks are not enough to cover all trips. The trips that are not covered are completed by hiring temporary workers or outsourced to external transporters.

1.2. Overview planning and scheduling

This section gives an overview of the differences and relations between the schedules and plans in the company. Figure 3 shows this overview. On the vertical side, the time units are given and on the horizontal side, the type of schedule is given. The last row contains the responsible person for each type of schedule. The text below the figure elaborates on the figure.

Time \ Type of schedule	Route plan	Block schedule	Driver schedule
6 to 12 weeks	Tactical route plan	Tactical block schedule	Base driver schedule
Week	Weekly route plan	Weekly block schedule	Weekly driver schedule
Day	Shift		
Hours	Trip		
Minutes	Stop		
Responsible person	**Supply Chain Planner**	**Transport specialist**	**Operational Transport Planner**

Figure 3: Overview planning & scheduling

The Supply Chain Planner creates the tactical route plan. The tactical route plan is a plan that indicates the forecasted amount and frequencies of deliveries for the stores. The tactical route plan is used for a certain period. This period varies between 6 and 12 weeks, depending on the time of the year. One week of the plan is called a 'weekly route plan'. This plan contains all trips to be completed in one week, already structured in shifts for drivers. A shift is defined as one working day for one driver, completed with one truck. Most shifts contains 2 or 3 trips. A trip is defined as a roundtrip that starts at the DC, the truck gets unloaded and ends again at the DC. Each trip consists of one or more stops. A stop is a place where the truck has to unload containers (and take back returns/packaging). In this situation, the truck has to unload containers at a store.

The block schedule is created by the transport specialist. The block schedule is not a separate plan, but a visualization of the tactical route plan. The block schedule is created for two purposes: planning routes and scheduling personnel. The scheduling of personnel is relevant for this research. The block schedule is printed on paper and used to base the driver schedule on. It is called block schedule, since the visualization is done by visualizing the trip in a rectangle/block. The visualization of the block schedule also contains the same shifts as the tactical plan, which are divided in trips with stops.

The operational transport planner uses the tactical plan as input for the driver schedule. Based on the tactical plan, the transport planner creates a base driver schedule. The base driver schedule is a schedule covering 6 working days. This base schedule is used during the period the tactical plan is also used. The real days a driver has to work, depends on the working days of the driver. Each driver has a roster indicating which days he or she has to work in each week. So, if a driver has to work 4 days in a week, the weekly schedule contains these 4 shifts of the base schedule. The weekly schedule consists of a shift on a day, which again consists of trips with stops assigned to the drivers.

Chapter 2 elaborates more on the responsible persons and terminology.

1.3. Research motivation

The main reason for this research is that the organization will implement a new advanced planning system (APS). A part of the APS is the route planning system, which is relevant for this research. The system will be implemented in the whole organization, so both the tactical and the operational route planning will be done within this system. An advantage of the new system is that it can indicate whether two or more trucks arrive at the same time at a store. In this way the planners can avoid waiting time for drivers.

Another advantage of the new system is the visualization of the routes. The production planners will see the routes they are planning on their screen. These routes will have a forecasted amount of containers. The system will indicate the fill rate of the trucks. When the orders are placed and differ from the forecast, Production Planning can re-plan these routes if needed. To add up on that, the waiting and transportation times are also visible on the screen. With each adjustment the planner can immediately see the result of his or her actions. The core task of planning routes will not change drastically, but the way it is executed is made easier. By visually seeing the routes, the company expects to better re-plan the routes.

Currently the planning of routes is mainly done on paper. The transport department creates the block schedule which is printed on paper. This printed version is used for re-planning the routes. This manually planning is very sensitive for failures and takes a lot of administrational effort. By the introduction of the new system, this paper block schedule will disappear.

An aspect that is not included in the new system, is the driver scheduling. So after the routes are planned, a schedule for the drivers still has to be made. Scheduling drivers is a time-consuming job, since a lot of the drivers have restrictions that have to be taken into account. These restrictions are related with health conditions, contractual requirements and preferences.

Because the new system will be implemented, the current way of working will not be maintainable. Therefore right now is the perfect moment to optimize the way of scheduling and to lower the labour intensiveness. Visualizing the tactical plan into the block schedule and then creating a weekly schedule for the drivers is a very time-consuming task and thus induces high personnel cost. Also since it is done manually, the process of scheduling drivers is very sensitive for failures.

1.4. Problem statement

This section gives the problem statements. Section 1.4.1 visualizes the problem and its causes by the use of a problem cluster. Section 1.4.2 elaborates on the cost of the schedule. The sections ends with Section 1.4.3 describing the problem owner.

1.4.1. Problem cluster

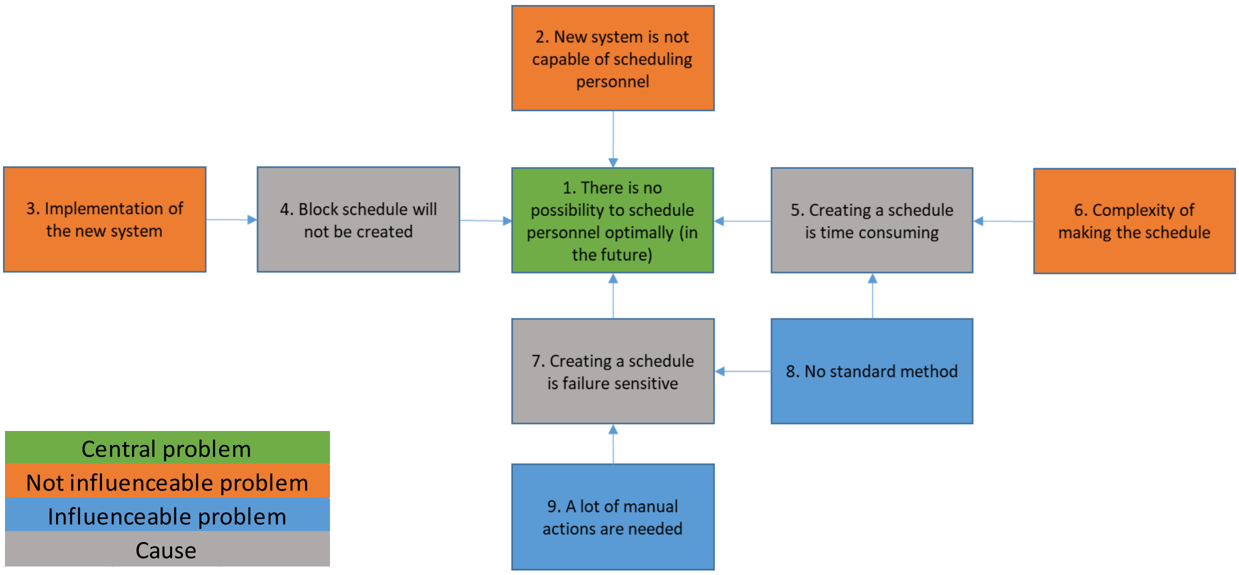

Figure 4: Problem Cluster

To come to the main cause of the problem, we use the problem cluster from the Managerial Problem Solving Method from Heerkens & Van Winden (2017). Figure 4 shows the problem cluster. We use the numbers in the boxes in the text below to link the text with the figure.

The central problem that exists is that (in the future) there is no possibility to schedule personnel optimally (1). This is also the initial problem given by the company in order to conduct this research.

New system is not capable of scheduling personnel

The first reason that there is no possibility to schedule personnel optimally is that the new APS is not capable of scheduling personnel (2). The new system will be implemented in the whole organization. The distribution center where the research is conducted encounters the problem of scheduling drivers the most, since they have the largest driver pool. The prioritization of scheduling personnel was not urgent enough to take into account for the whole organization. This means that the distribution center has to come up with a solution itself.

Block schedule will not be available

The second reason that there is no possibility to schedule personnel optimally, is that the block schedule will not be created anymore (4). Since the routes will be planned within the new system, the route planning on paper disappears. So due to the implementation of the new system, the need for the block schedule also disappears. Since the block schedule is mainly built to facilitate the route planning on paper, the company has decided to get rid of the block schedule when the new system is implemented (3). Also, the visualization is done in the weekends by the transport specialist. The transport specialist is the only employee which executes the visualization. In case of emergency, other employees can make the block schedule. However, this will take a significant amount of time. This results in that the process of visualizing is a process with a high risk, since no 'good' fall-back exist.

Creating a schedule is time consuming

The third cause that there is no possibility to schedule personnel optimally in the future, is that the process is time consuming (5). That the process is time consuming is caused by the complexity of making the schedule (6). The scheduling process is complex since the operational transport planners have to take several aspects into account: high utilization of own trucks, agreements with external transporters, medical conditions and contractual requirements of the drivers. Also, the absence of a standard method (8) influences the time spent to create a schedule.

Creating a schedule is failure sensitive

The fourth reason that there is no possibility to schedule personnel optimally, is that the current method of creating a schedule is failure sensitive (7). For example if a constraints is not satisfied, this results in an infeasible schedule. It can happen that constraints are not satisfied by the planner due to the high failure sensitivity of the process. The failure sensitivity is caused by that a lot of manual actions are needed to create a driver schedule (5). Also, the absence of a standard method (10) influences the failure sensitivity of the process.

The independent problems that have influence on that there is no possibility to schedule personnel optimally are: a lot of manual actions are needed and there does not exist a standard method. These are the core influenceable problems that are solved by the result of this research.

1.4.2. Cost related to the schedule

There are costs associated with the creation of the schedule and the schedule itself. The first cost related to the schedule is the time it takes to create a schedule. It takes time to create a driver schedule taking into account all restrictions. This time is expressed in hours worked by the operational transport planner. This cost for the creation of the schedule can be calculated by the hours spent times the hourly wage.

The second aspect is the cost regarding the quality of the schedule itself. One can make a schedule in a short time, but which is probably of poor quality. A good schedule has the aspects of a low number of undertime and overtime hours. A poor quality schedule results in higher cost than a good quality schedule when being executed.

- Undertime hours are hours that have to be paid due to contractual hours, but are not worked. The hours that are not worked by the company drivers are indirectly assigned to temporary workers or external transporters.
- Overtime hours are hours that are worked on top of the contractual hours. These hours are paid with a certain overtime percentage.

Undertime hours are ‘paid twice’ since the company drivers are paid and temporary workers or external transporters are paid as well. Overtime hours are paid with a certain overtime percentage, the total hourly rate of overtime hours is more expensive than the rate of the external transporters. Fluctuations in the working hours per week are also not desirable regarding the satisfaction of the employees. The contractual hours are agreed hours for the drivers and big deviations on a frequent base is not desired. So, in general for each driver undertime and overtime hours should be avoided. However, situations can exist where allowing overtime hours can minimize the total cost, for example where using overtime avoids hiring extra people.

1.4.3. Problem owner

The problem owner in this research is the transport manager. He owns the problem that there is a discrepancy between norm and reality (Heerkens & Van Winden, 2017). In this research, the norm is that at least the same quality schedule for drivers can easily be created without failures. Easily is defined as half of the effort it now takes to create a schedule. The reality currently is that creating a weekly schedule takes a lot of time and is very sensitive for failures. The transport manager is also responsible for the budget regarding transport. So, bad quality schedules that result in high transport cost affect his performance.

1.5. Research objective and scope

This section describes the goal and frame of the research. Section 1.5.1 describes the research objective and Section 1.5.2 defines the scope of the research.

1.5.1. Objective

Based on the problem cluster that is described in the previous section, the research objective is formulated as follows:

> *“To find a method that creates a feasible/near-optimal driver schedule with the lowest cost possible.”*

The objective of this research is to develop a method that can be used in the future to create a (near-optimal) weekly schedule for drivers. The method should be capable of taking restrictions of the drivers into account.

1.5.2. Scope

The scope of this project is creating a schedule with the given restrictions. These restrictions cannot be changed in this research. Also the implementation of the new advanced planning system is not influenceable by this research.

The tactical plan is input for this research. Outside of this tactical plan, there are more trips that have to be completed. These trips are out of scope. Also the content of the tactical plan cannot be changed. Both the tactical route plan and the weekly route plan are already structured in shifts. One shift can consists of multiple trips. These shifts stay intact.

Finally the result is demarcated to the weekly schedule. This means that operational/daily scheduling is not taken into account. These fluctuations in the schedule are for the responsibility of the operational transport planners.

1.6 Research design

To translate the problem described above and achieve the research goal that is formulated, we formulate research questions that have to be answered. First the main research question is formulated. The main research question is divided into sub questions in order to structure the research. These sub questions are given below the main research question. Each sub question represents a chapter in this report. The outline of the report is given by describing the content of each chapter below each question.

Main research question:

> ***"What method should the company use to create a weekly schedule for its drivers that can be used along with the advanced planning system with the aim of lowest cost possible?"***

Sub research questions:

1. How is the current process of scheduling drivers organized and how does it perform?

In order to come up with improvements, Chapter 2 gives a detailed description of the current situation. We describe what steps are currently executed in order to visualize and create a schedule for the drivers. In Chapter 2 we also elaborate on the schedule restrictions mentioned earlier. Also, an analysis on the current performance of the current situation is given. To sketch the complete current situation, observations are made and interviews are conducted with the Supply Chain Planner and the operational transport planners.

2. What theory and methods exist in literature to improve scheduling personnel?

Chapter 3 provides a literature review about relevant literature for the research. Literature about scheduling and rostering is reviewed. We look into problem characterizations and formulations in order to give a complete description of the problem. Next to that, we discuss optimization techniques to solve the problem.

3. How to build a weekly driver schedule for the company with the aim of lowest cost possible?

Chapter 4 contains the design of the solution. This is done by applying aspects from different literature sources described in Chapter 3 to this specific problem. We formulate the problem of the current situation as done in literature taking into account all aspects that are mentioned in Chapter 2.

4. How does the method perform (compared to the current situation)?

In order to test the performance of the method, we generate and test different instances in Chapter 5. We also analyse how the proposed method performs under different conditions. In order to analyse the outcome, we first define different key performance indicators that can be used to score the quality of a schedule.

Chapter 6 contains the conclusions and recommendations. We already discuss the use of the method in practice. In order to make it usable for the company, we take into account different aspects that are perceived as critical success factors during implementation. Furthermore, the chapter gives suggestions for further research.

2. Context analysis

This chapter answers the first research question stated in Section 1.6: *"How is the current process of scheduling drivers organized and how does it perform?"*

This chapter starts with elaborating on some terminology regarding the scheduling and rostering in Section 2.1. Section 2.2 describes the current process of scheduling drivers. Section 2.3 gives characteristics and an indication of the size of the problem. Section 2.4 describes the performance of the current situation. The chapter finalizes with the conclusions in Section 2.5.

2.1. Terminology

This section describes the terminology used within the company and the project. Section 2.1.1 gives an overview of the terms and their relations. Sections 2.1.2 and 2.1.3 elaborate on the tactical plan and the block schedule, respectively. The section ends with Section 2.1.4 describing the base driver schedule.

2.1.1. Overview

To give clear understanding of the current situation, we first introduce aspects and relations regarding the route and driver schedule. Chapter 1 already introduced Figure 5, but we elaborate more on it in the next subsections.

Time \ Type of schedule	Route plan	Block schedule	Driver schedule
6 to 12 weeks	Tactical route plan	Tactical block schedule	Base driver schedule
Week	Weekly route plan	Weekly block schedule	Weekly driver schedule
Day	Shift		
Hours	Trip		
Minutes	Stop		
Responsible person	**Supply Chain Planner**	**Transport specialist**	**Operational Transport Planner**

Figure 5: Overview planning & scheduling

2.1.2. Tactical plan

The scheduling process for the drivers starts with the tactical plan. The tactical plan is a plan with a planning horizon of a week that contains all trips to supply the stores. This plan has a planning horizon of a week, but is valid for a specific number of weeks (usually between 6 and 12). The tactical plan is composed by the Supply Chain Planner. The Supply Chain Planner uses a forecasted number of orders and time windows of stores as input to create the tactical plan.

The Supply Chain Planner creates shifts for the drivers. A shift is a working day for a driver and can consist of multiple trips. Each trip consists of one or multiple stores that are supplied. When creating the tactical plan, the Supply Chain Planner takes different aspects into account: minimum and maximum length of a shift, at most 3 trips per shift, number of company trucks available, different types of trailers and agreements with external transporters. The planner uses the time windows from the stores as input and tries to make as optimal shifts as possible. By only using the time windows, the start times of each trip may vary. This means that the departure time of trucks varies. When the departure times vary, this also means that the start times of the drivers vary within the week.

Figure 6 shows a part of a tactical plan. Each row in this plan represents an activity to be executed. An activity can be unloading containers or loading returns/packaging. The complete tactical plan is a list of around 5000 rows, depending on how busy it is during that time of the year. A shift can be recognized by the column 'truck'. The number of trips can be found back in the column 'route number', where route is another term for trip. The list contains the start time, end time and total time of the shift. The type of activity can be found back in the column ‘product’. DKW1 means non-fresh products, VERS and CCJ mean that the truck contains fresh products and EMB1 is for returning containers and packaging. The returning of containers and packaging is included in a trip and is done after the drivers unloads the truck. However, this activity is separately mentioned in the route planning. The type of activities that have to be executed determine the type of shift.

Day	Route number	Customer	Customer name	Unload	Load	Product	Activity number	Departure time DC	Arrival at customer	Return DC	Order number	Truck	Trailer	Driver	Shift time	Kilometers	Product type	Start shift	End shift
1	1	x	Stop 1	30		VERS	1	10:24	12:05	15:32	x	1	x	x	9:39	296 KM	K	9:38	15:55
1	1	x	Stop 1	18		DKW1	2	10:24	12:05	15:32	x	1	x	x	9:39	296 KM	D	9:38	15:55
1	1	x	Stop 1		48	EMB1	3	10:24	12:05	15:32	x	1	x	x	9:39	296 KM	EMB 48	9:38	15:55
1	2	x	Stop 1	19		VERS	1	16:41	17:29	19:35	x	1	x	x	9:39	296 KM	K	9:38	19:58
1	2	x	Stop 1	1		CCJ	2	16:41	17:29	19:35	x	1	x	x	9:39	296 KM	C	9:38	19:58
1	2	x	Stop 1	29		DKW1	3	16:41	17:29	19:35	x	1	x	x	9:39	296 KM	D	9:38	19:58
1	2	x	Stop 1		48	EMB1	4	16:41	17:29	19:35	x	1	x	x	9:39	296 KM	EMB 48	9:38	19:58
1	3	x	Stop 1	24		VERS	1	10:56	12:08	15:08	x	2	x	x	9:31	353 KM	K	10:08	15:31
1	3	x	Stop 1	1		CCJ	2	10:56	12:08	15:08	x	2	x	x	9:31	353 KM	C	10:08	15:31
1	3	x	Stop 1	25		DKW1	3	10:56	12:08	15:08	x	2	x	x	9:31	353 KM	D	10:08	15:31
1	3	x	Stop 1		48	EMB1	4	10:56	12:08	15:08	x	2	x	x	9:31	353 KM	EMB 48	10:08	15:31
1	4	x	Stop 1	11		VERS	1	15:59	17:17	19:57	x	2	x	x	9:31	353 KM	K	10:08	20:20
1	4	x	Stop 1	1		CCJ	2	15:59	17:17	19:57	x	2	x	x	9:31	353 KM	C	10:08	20:20
1	4	x	Stop 1	21		VERS	3	15:59	17:41	19:57	x	2	x	x	9:31	353 KM	K	10:08	20:20
1	4	x	Stop 1		48	EMB1	4	15:59	17:41	19:57	x	2	x	x	9:31	353 KM	EMB 48	10:08	20:20

Figure 6: Tactical route plan

2.1.3. Block schedule

The Supply Chain Planner delivers a complete tactical plan in Excel. However, this Excel-list is not workable for the distribution center yet. In order to make it workable for both the production planners and the operational transport planners, the plan is visualized. The transport specialist visualizes the tactical plan into the block schedule.

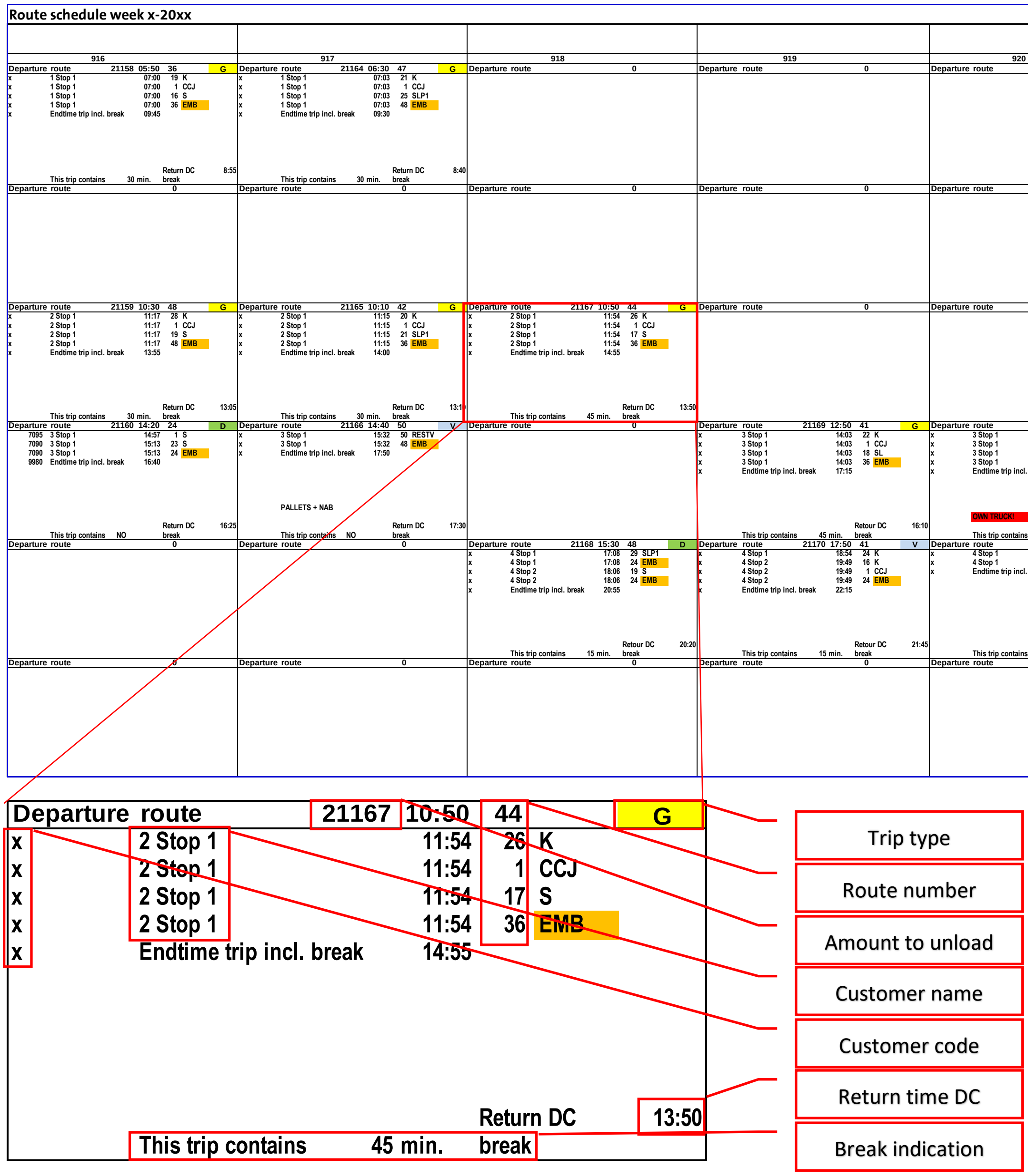

Figure 7: Block schedule

Figure 7Figure 7 shows an example of the block schedule that is currently used. The names of the stops are left out due to anonymity of the company. The figure shows only one page of the whole schedule. Each column represents a shift and each block represents a trip. So, the figure shows 5 shifts with 2 or 3 trips per shift. The position of the block corresponds with the moment of delivery for the stores, so the higher the block, the earlier the moment of delivery.

The figure zooms in on one block to show what information the block contains. One block represents one trip and contains information about stores to deliver, how much time is assigned to the trip, eventual breaks and what type the trip is. There exist three types of trips: 'D', 'V' and 'G'. 'D' stands for 'DKW', which means non-fresh products. 'V' stands for 'Vers', which indicates fresh products. The third type of trip is indicated with a 'G', which stands for 'Gemengd'. 'Gemengd' means that there is a mix of both fresh and non-fresh products in the truck. For the remainder of the report, we use 'mixed' as an indication for 'Gemengd'. Furthermore, the block schedule contains important information about trips. For example, some trips need their own truck or a truck contains pallets instead of containers; this information is also included in the block schedule.

It is important to note that the transformation from the tactical plan to the block schedule is only a visualization task. Neither new information is added nor decisions are made in the process. The visualization task is done by the transport specialist. This is the only person in the whole company that performs the visualization regularly. This means that the process is a high-risk process since there is no good fall-back if the transport specialist cannot do this anymore.

The estimation of the time the transport specialist needs to make the first version of the block schedule is around 5 hours. When the first version is done, an operational transport planner and a production planner perform a check on this version. When these checks are done, the version is finalized with the given feedback. The finalization of the schedule takes approximately 3 hours. So, the complete build-up of the block schedule takes around 8 hours.

2.1.4. Base driver schedule

The base driver schedule is a schedule for each driver, usually covering 6 days per week per driver. Sometimes the base schedule covers 4 or 5 shifts, depending on contractual agreements for free days. The base schedule is created to make weekly schedules, taking into account free days and sickness. Each driver has a yearly roster indicating which days he or she has to work each week. These working days are filled with shifts from the base schedule to create a weekly schedule. The shifts that are left over, are executed by external transporters or temporary workers.

Figure 8 shows how the company applies the base driver schedule. On the left hand the driver number and name are indicated. The first columns of MTWTFSS contains the block numbers that can be found back in the block schedule. Each block number represents a shift. The colours of the blocks indicate the type of shift. On the right hand the MTWTFSS columns contain the shift lengths corresponding to the block numbers on the left hand.

Driver No.	Driver name	M	T	W	T	F	S	S	Hours	M	T	W	T	F	S	S
13	Driver name	205	205		210	215	210		47:30	9:00	10:05		9:20	9:35	9:30	
16	Driver name	922	401	901	905	936	928		57:15	10:15	9:10	10:00	9:10	9:20	9:20	
14	Driver name	926	905	922	930	910	202		56:20	10:20	8:30	9:05	9:05	9:30	9:50	
118	Driver name	944	941	943		951	952		49:35	9:35	10:15	10:10		10:00	9:35	
27	Driver name	940	915	909	928	405	945		64:45	11:20	10:25	10:00	10:55	11:15	10:50	
30	Driver name	928	923	920	908		910		50:25	10:05	10:25	9:40	10:40		9:35	
31	Driver name	914	902	902	204	905	935		60:45	10:40	10:40	11:05	10:25	8:50	9:05	
32	Driver name	918	951		903	901	902		51:20	10:05	9:20		11:00	10:20	10:35	
45	Driver name	943	932	944	953	952	926		56:55	9:25	10:05	9:05	9:35	9:30	9:15	
85	Driver name	930	929	926	912	941	937		59:25	10:10	9:25	10:15	10:00	10:25	9:10	
46	Driver name	941	946	906	946	956	942		62:15	10:25	9:50	10:40	10:15	10:40	10:25	
54	Driver name	933	948	907	926	928	207		57:15	8:40	10:00	9:40	9:15	9:35	10:05	
43	Driver name	932	907	946	949	949	947		58:20	9:10	10:00	9:20	10:10	9:40	10:00	
81	Driver name	927	928	928	911	943	911		59:55	10:00	10:25	10:00	9:35	10:25	9:30	
93	Driver name	203	204	203	213		208		51:15	10:10	11:55	9:40	9:40		9:50	
7	Driver name	207	206		211	203	209		50:55	10:25	10:25		10:05	10:10	9:50	
49	Driver name	905	914	929	942	947	912		56:00	9:00	9:50	8:50	9:10	9:30	9:40	
108	Driver name	401	926	927	937	940	938		59:15	9:10	10:20	9:25	10:10	9:45	10:25	
68	Driver name	204	207	204	208	214	211		64:00	10:15	11:30	10:25	11:15	10:10	10:25	
18	Driver name	912	936	937	919		944		49:10	9:55	10:30	8:50	9:50		10:05	
39	Driver name	929	904	925	939	912	936		60:10	10:05	9:15	10:35	9:50	10:00	10:25	
15	Driver name	921	927	919	403	204	930		57:20	9:40	9:15	9:20	10:00	9:35	9:30	
115	Driver name	935	931	951		213	922		59:00	11:35	11:40	11:45		12:00	12:00	
121	Driver name	946	940		941	927	940		48:00	10:00	10:00		9:05	9:45	9:10	
47	Driver name	902	921		933	906			37:45	9:15	9:55		8:55	9:40		

Non-fresh	Mixed	Fresh	Own truck	Pallets	Standby

Figure 8: Base driver schedule

2.2. Scheduling drivers

This section describes the process of scheduling drivers. Section 2.2.1 describes the process, Section 2.2.2 elaborates on the restrictions of the drivers. Sections 2.2.3 and 2.2.4 describe the temporary workers and external transporters, respectively.

2.2.1. The process

Figure 9 gives the complete process of creating a weekly schedule for the company drivers, temporary workers and external transporters.

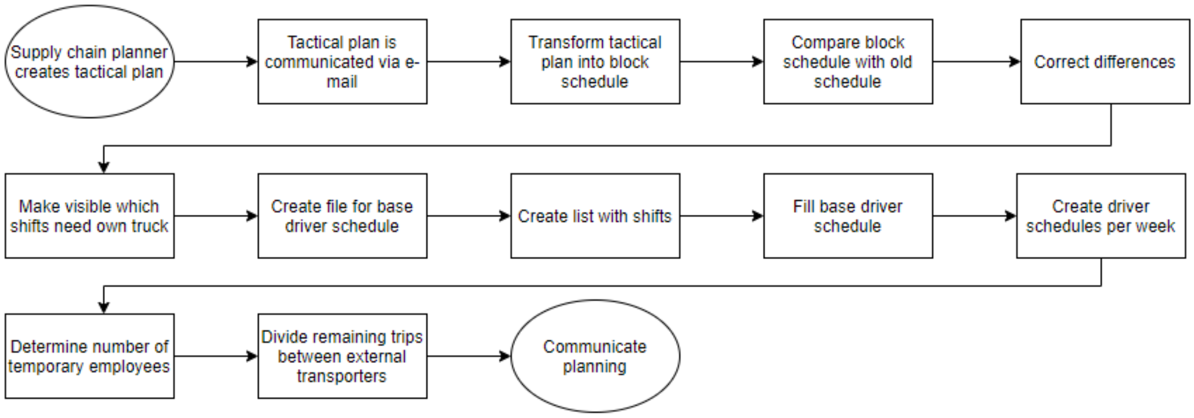

Figure 9: Process of creating the driver schedule

The first step in the driver scheduling process is that the Supply Chain Planner creates the tactical plan, by using forecasts and taking time windows into account. When the tactical plan is made, the Supply Chain Planner communicates the plan via e-mail with the operational transport planners.

When the tactical plan is known in the distribution center, the transport specialist transforms the plan into the block schedule. The complete build-up of the block schedule takes around 8 hours.

The operational transport planners compare the new block schedule with the old schedule. The differences are written down and adjusted to know what changed between the new and old schedule. Some shifts need their own truck due to the height of the truck or the specific cooling system in the trailer of the own truck. This is the reason that the operational transport planners indicate which shifts need their own truck.

When all information needed to create the driver schedule is gathered, a file for the base driver schedule is created. Next to that, the operational transport planner generates a list with shifts to be assigned. These shifts are assigned to the drivers to create the base driver schedule. The base driver schedule is a base schedule usually covering 6 days. From this base schedule, weekly schedules are made taking into account free days and sickness.

When the schedule for the company drivers is finished, the leftover trips are assigned to temporary employees and external transporters. The number of temporary employees depend on the number of company trucks available. The external transporters execute the remaining trips. Finally, the weekly schedule is communicated to the drivers, employment agency and external transporters. Communication to the company drivers is done by the use of printed papers. Figure 10 shows an example of such a printed paper.

Week schedule Transport

Driver:	Driver's name					**Week:**	**x**	
Driver.nr. :	xx							
Day	**Block**	**Trip1**	**Trip2**	**Trip3**	**Trip4**	**Start**	**End**	**Total Time**
MONDAY	910	21134	21135	21136		06:20	17:45	11:25
TUESDAY								
WEDNESDAY	911	23169	23170	23171		06:30	17:45	11:15
THURSDAY	910	24090	24091			07:45	17:55	10:10
FRIDAY	942	25106	25107			07:55	18:45	10:50
SATURDAY								
SUNDAY								
TOTAL								**43:40**

Figure 10: Paper schedule per driver

The complete process is done in Excel and on paper. Also, all steps in the process are done manually. This results in that the process is time-consuming and very sensitive for failures.

2.2.2. Restrictions of the drivers

The restrictions of the drivers are aspects that add a lot of complexity to the schedule. Some restrictions are prescriptions of the doctor due to the medical condition of the drivers, some are agreements made in the past and others are preferences that are taken into account. A distinction can be made between hard and soft restrictions. Hard restrictions have to be met, soft restrictions are desirable to meet. There are a few types of restrictions that have to be taken into account. These restrictions are:

Working days of the driver

The days drivers have to work in each week are given in the yearly schedule. This schedule is generated before the start of the year and is made to make an even spread within the weeks. Also the number of Saturdays drivers have to work is divided fairly.

Working hours per week per driver

The number of working hours per week is given in the contract of the drivers. There are some differences between drivers in contractual hours. These differences descend from historical agreements. It is the task of the operational transport planners to come approximately to the number of working hours per week of the drivers. Hours above the contractual hours have to be paid out with an overtime percentage. Hours below the contractual hours that are not worked, but still have to be paid.

Start time/end time

Another restriction that has to be taken into account is the start and end times of the shift. Some drivers have agreements or preferences regarding the start or end time. For example, some drivers prefer to start earlier than others.

A legal requirement regarding the start time is the maximum deviation in start time between consecutive days. So, between consecutive days, there cannot be more than 2 hours deviation between the start times.

Total duration of the shift

The total duration of the shift a driver has to work can also differ between drivers. Usually, shifts for drivers are long shifts of over 10 hours of working time. To comply with a working week of 40 hours, the drivers usually work 4 days. However, some drivers prefer to work 8 hours per day for 5 days.

Type of shift

With the type of shift it is meant what types of products are shipped. As mentioned earlier, 2 types of products are shipped. These shipments of products result in 3 types of shifts, namely non-fresh, fresh or mixed (mixed is a combination of non-fresh and fresh). Due to medical conditions, some drivers cannot execute fresh or mixed shifts.

Number of trips per shift

The last restriction is the number of trips per shift. The average age of the drivers is relatively high (around 58 years old). This results in the concern that the drivers' physical condition can decline when putting too much physical stress on them. Each trip means extra loading and unloading, so the more trips the more physical load. So, some drivers have agreed to have a maximum of 2 trips per shift due to medical conditions. To add up on that, the number of trips per shift should be evenly spread over the drivers. So if a driver works 4 days, he should ideally have 2 shifts with 2 trips and 2 shifts with 3 trips.

2.2.3. Temporary workers

The company uses temporary workers to have a flexible working pool. Temporary workers do not have to meet a required number of hours per week. The temporary workers are paid per hour and make use of the trucks owned by the company. The advantage of using temporary workers is that they are easy to scale up and down. Temporary workers are scaled up and intensely used in holiday periods. Scaling down is needed in periods where all own drivers are working and thus almost all company trucks are occupied.

The main reason that the company makes use of temporary workers is the utilization of their own trucks. Otherwise the trucks would be underutilized, since most own drivers only work 4 days per week.

2.2.4. External transporters

Since the own pool of drivers and trucks is insufficient to cover all trips, external transporters are used. External transporters are, contrary to temporary workers, using their own trucks to complete the trips. To reduce the cost as much as possible, there are some agreements made with the transporters. The agreements that are made include aspects like the total hours per truck, what trailer they use, etc. Also the external transporters are easy to scale up and down and used to overcome fluctuations over the week.

2.3. Characteristics

To give an indication of the size of the problem, this section describes characteristics of the route plan. The distribution center is a regional distribution center. It has to supply fresh and non-fresh products to 160 of the around 700 stores. Each store has its own time windows during which the products can be delivered. These time windows were assigned to the stores in the past. Some stores are bigger than others and trucks sometimes arrive more than once a day. Other stores are smaller and receive a few deliveries per week. Figure 11 shows the number of trips per day.

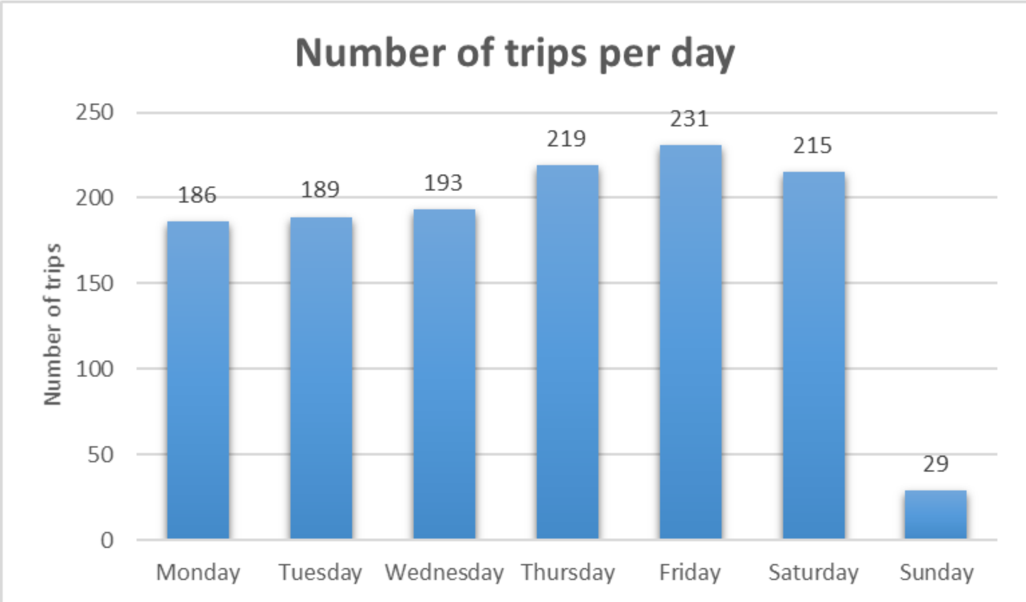

Figure 11: Trips per weekday

In order to complete all these trips, the earlier mentioned tactical plan is created. The tactical plan consists of shifts with multiple trips in each shift. Usually a shift contains 2 or 3 trips, depending on the truck load and the distance to the stores. Figure 12 shows the total number of shifts per day.

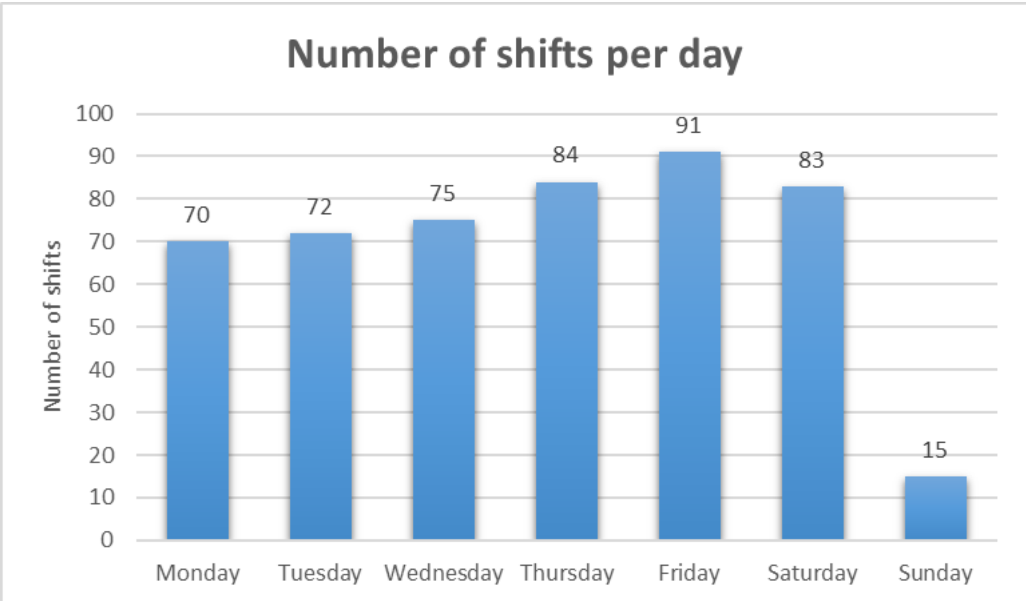

Figure 12: Shifts per weekday

The workload of a shift within the company is defined by the number of trips per shift. So the more trips a shift has, the higher the workload. The average amount of trips per shift throughout the week, shows little differences. Figure 13 presents the average trips per shift per weekday.

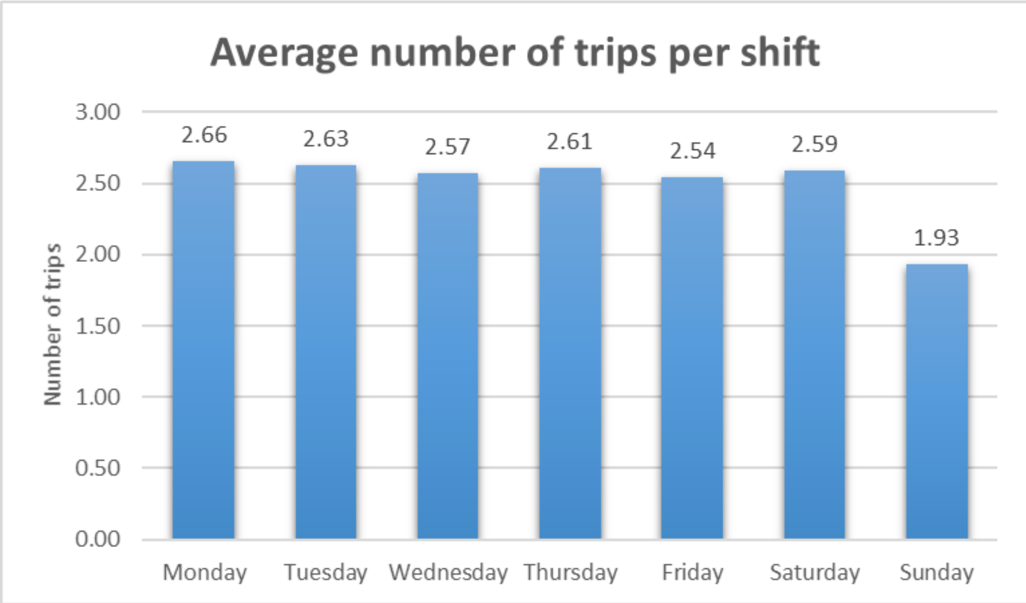

Figure 13: Average trips per shift per weekday

The final characteristic is the average length of a shift. Figure 14Figure 14 presents the average shift length per weekday. The length of a shift is the total working day of the driver, including breaks.

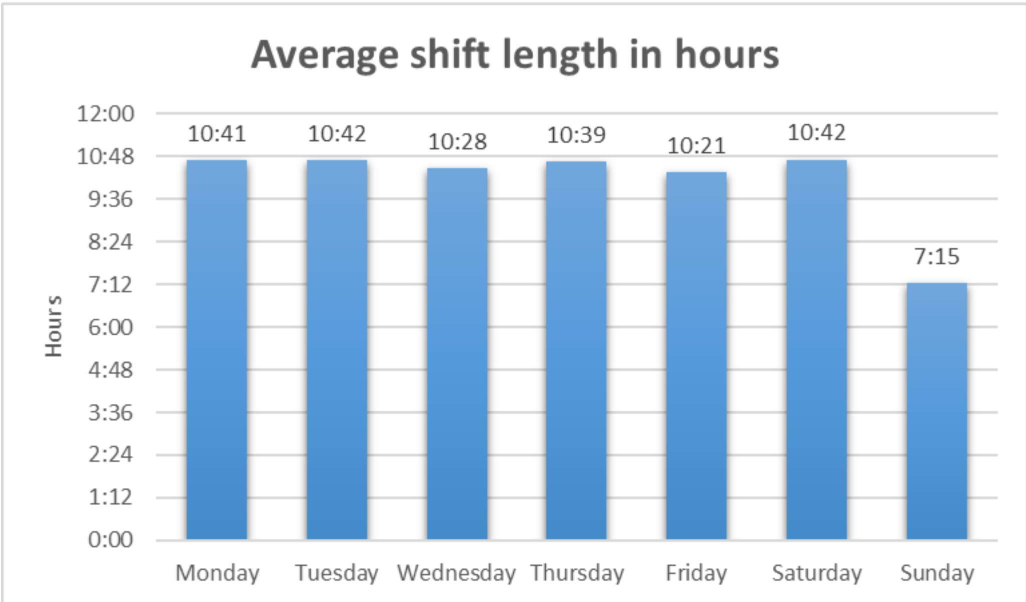

Figure 14: Average shift length per weekday

2.4. Current performance/ Key performance indicators

This section discusses the current performance of the driver scheduling process. Section 2.4.1 elaborates on the Key Performance Indicators used in the company. Section 2.4.2 and 2.4.3 describe the quality of the schedule and deployment of the drivers, respectively. Section 2.4.4 elaborates on the effort it takes to create a schedule.

2.4.1. Key performance indicators

Currently, the company does not use Key Performance Indicators (KPIs) to measure the quality of the schedule. The company does use an indicator to review how the execution of the schedule went (schedule compliance). They compare the hours that are planned versus the hours that are used to complete the schedule. However, since the tactical schedule is our concern, the schedule compliance is out of scope for this project. The company unfortunately does not measure the quality of the schedule itself. When the transport planners are scheduling their drivers, they have to take into account some guidelines. When a schedule is created that satisfies the restrictions of the drivers, the schedule is completed. However, the quality of the schedule can be analysed, the next subsection discusses the quality of the schedule.

2.4.2. Quality of the schedule

To develop a proper measure of the quality of the schedule, we analyse schedules that are created in the past. We compare the scheduled hours to the contractual hours. Week 2 to 12 of 2021 are used for this analysis. Week 2 to 12 are 'normal weeks', which means that there are no special days in these weeks. However, the data of these weeks is not ready to compare. In order to make it useable, we filter the data. We filter out special cases from employees and part-time employees with deviating hours. The analysis shows the overtime, undertime and absolute difference.

The following formula gives the calculation for the overtime hours:

$$Overtime\ hours = MAX\big(0, scheduled\ hours - (contractual\ hours + break\ hours)\big)$$

The undertime hours can be calculated with the following formula:

$$Undertime\ hours = MAX(0, (contractual\ hours + break\ hours) - \ scheduled\ hours)$$

The absolute difference is calculated with the following formula:

$$Absolute\ difference = |scheduled\ hours - (contractual\ hours + break\ hours)|$$

The break hours are added since the breaks are included in the scheduled hours but contractual hours are expressed without breaks. After filtering the data, we show the results by the use of following box and whisker plots. A short description of the box and whiskers is given in Figure 15.

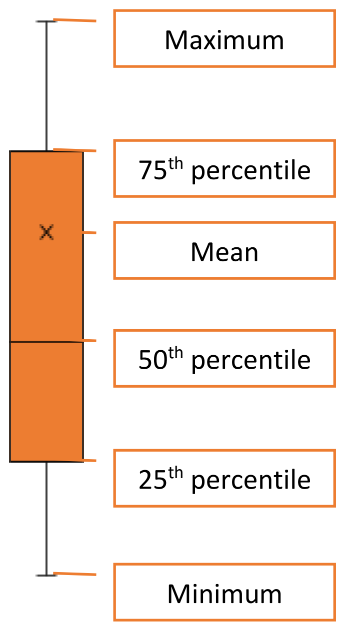

Figure 15: Box and whisker

Figure 16 shows the overtime hours per week:

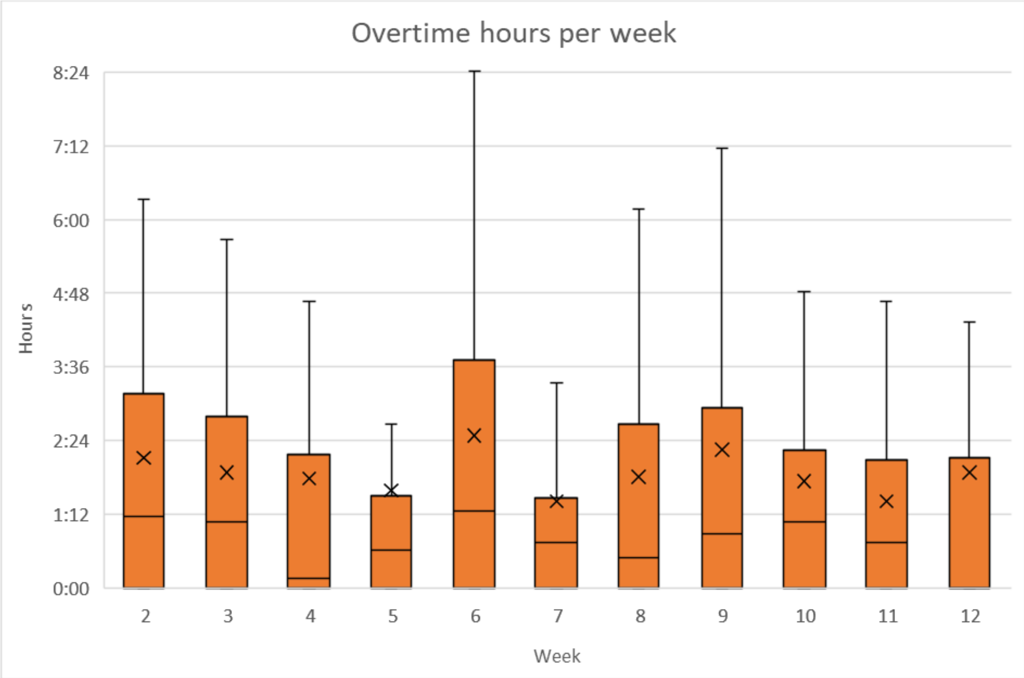

Figure 16: Overtime hours per week

Figure 17 shows the undertime hours per week:

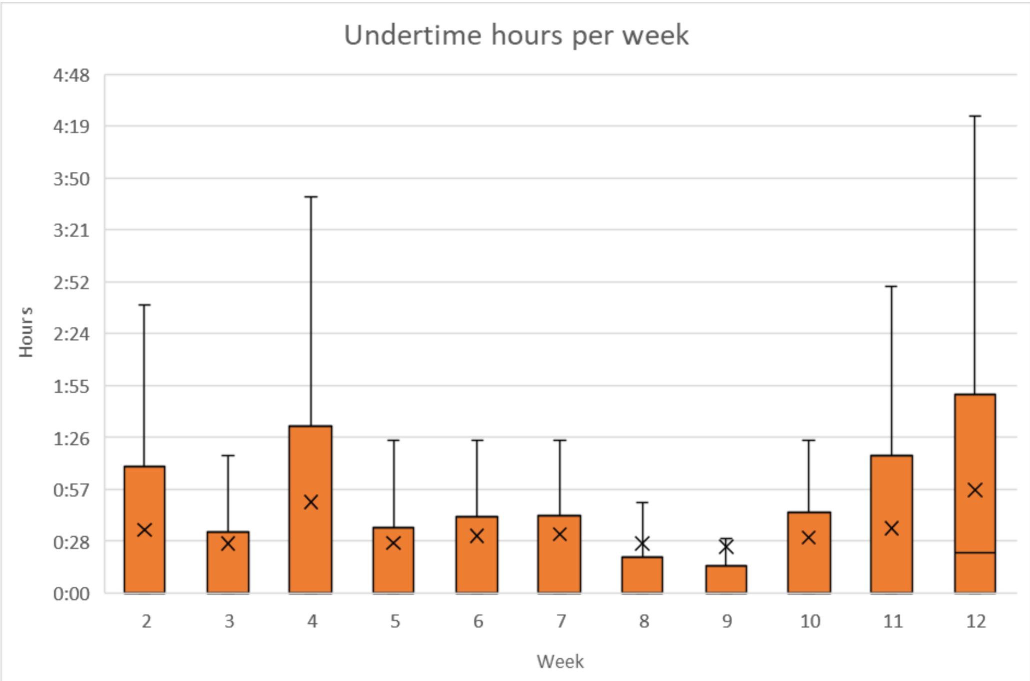

Figure 17: Undertime hours per week

Figure 18 contains the absolute difference in hours per week:

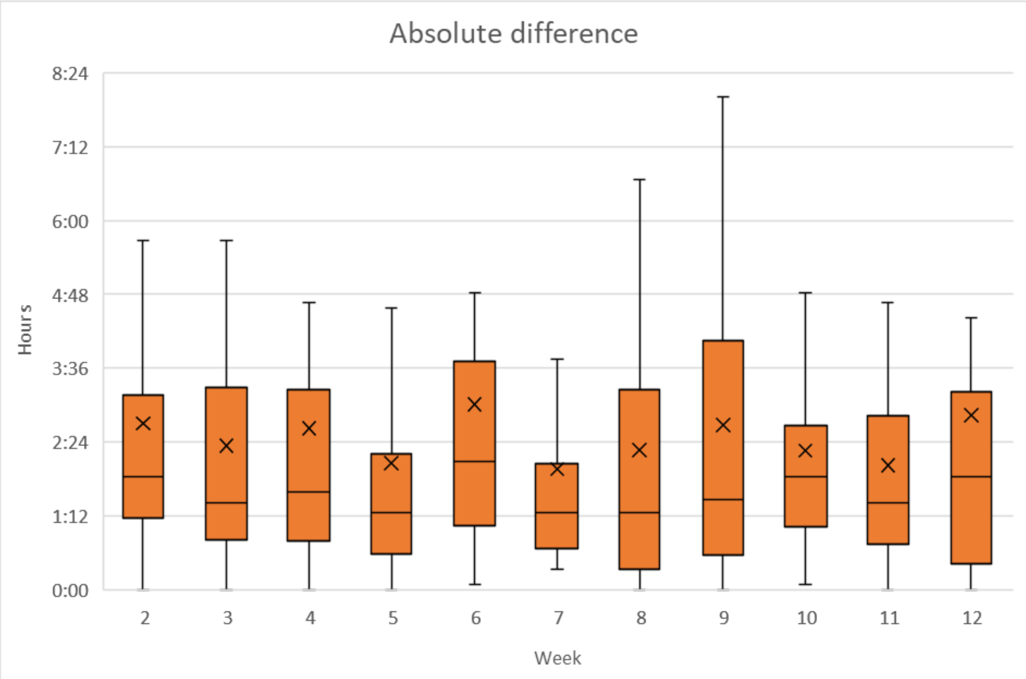

Figure 18: Absolute difference between contractual and scheduled hours

The box-and-whisker plots do not include outliers. Figure 16 shows that the average overtime is 1 hour and 51 minutes, Figure 17 shows that the average undertime is 34 minutes and Figure 18 shows that the absolute difference is 2 hours and 26 minutes. The outcome of the analysis shows that overtime hours are more common than undertime hours. The absolute difference of 2 hours and 26 seconds is relatively high, when keeping in mind that drivers work 36 or 40 hours per week. Also, the differences between scheduled and contractual hours are on a tactical level, so differences in time when executing the schedule are not taken into account here. Table 1 shows the overtime, undertime and the absolute difference per week schematically.

Week	Overtime	Undertime	Absolute difference
2	02:07:06	00:35:16	02:42:21
3	01:53:06	00:27:32	02:20:37
4	01:47:10	00:50:34	02:37:44
5	01:35:29	00:28:00	02:03:29
6	02:28:55	00:31:55	03:00:51
7	01:25:00	00:32:44	01:57:44
8	01:48:35	00:27:43	02:16:19
9	02:15:00	00:25:52	02:40:52
10	01:44:43	00:30:53	02:15:36
11	01:25:00	00:36:11	02:01:11
12	01:52:46	00:57:16	02:50:02
Average	**01:51:24**	**00:34:41**	**02:26:05**

Table 1: Overtime, undertime and absolute difference schematically

2.4.3. Deployment of drivers

The breakdown of the schedules can show what number of shifts are executed by company drivers, temporary workers or external transporters. Figure 19 shows this breakdown. On the horizontal side, the weekdays are given and on the vertical axis the type of driver is given. The analysis uses Week 2 to 12 of 2021. The last row in the figure shows the average use of own trucks. The deviation in own trucks is caused by sickness of employees or maintenance of the trucks. The number of own trucks is determined by adding the company drivers and the temporary workers since these make use of the company trucks.

Days / Type of driver	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Company drivers	39.9	40.5	39.9	38.1	37.1	20.5
Temporary workers	8.9	8.4	11.5	8.7	11.5	30.2
External transporters	34.6	38.0	35.0	50.5	52.6	43.4
Total	83.5	86.8	86.4	97.3	101.3	94.0
Used trucks	48.8	48.8	51.4	46.8	48.6	50.6

Figure 19: Deployment of drivers per weekday

The figure contains some noteworthy items. The first remarkable item is that Friday is the day with the most drivers needed, but has the second to last company drivers scheduled. This is caused by agreements arising from past contracts. The figure also shows that the peak moment for temporary workers is on Saturday. This is also caused by a low number of company drivers working this day.

2.4.4. Effort to create a schedule

It is also possible to make an estimation on the effort it takes to create a schedule. So it takes time to rebuild the tactical plan into the block schedule and sequentially translate the block schedule into a schedule for drivers.

There are 2 main tasks for creating the weekly schedule: creating the block schedule and scheduling the drivers. The transformation into the block schedule takes approximately 8 hours. The second task is assigning drivers to the shifts. To build a weekly schedule for drivers, approximately 40 hours are needed. So the total hours that are needed to create the weekly schedule is 48 hours (8 + 40).

Besides the time it takes to create the driver schedule, the process is also very sensitive for failures. This is due to the manual work in the process of scheduling drivers. However the failure sensitivity is hard to quantify and the results of the failures are not documented by the company.

2.5. Conclusions

This chapter answered the following research question: *"How is the current process of scheduling drivers organized and how does it perform?"*

To summarize, the current process for creating a weekly schedule is done by converting the tactical plan into the block schedule and transform this into a base schedule for drivers. The weekly schedule is created from this base schedule with free days, sickness and other aspects taken into account. The weekly schedule is communicated to the truck drivers, employment agency and external transporters. The restrictions of the drivers and the manual work are the reasons that the process is a very time-consuming and failure sensitive process.

Furthermore, the company does not use key performance indicators to measure the performance of generated schedules. However, the outcome of our analysis showed an average undertime of 34 minutes, overtime of 1 hour and 51 minutes and an average absolute deviation of 2 hours and 26 minutes between the scheduled and contractual hours. Section 2.4.3 showed the deployment of the drivers and the number of own trucks used. The total time that was needed to create a weekly schedule from scratch is 48 hours.

3. Literature review

This chapter gives an overview of relevant literature that is available about scheduling and rostering. Here we answer research question 2: *“What theory and methods exist in literature to improve scheduling personnel?”*

This chapter starts with an introduction to scheduling and rostering in Section 3.1. Section 3.2 describes specific types of scheduling problems in different industries that are somewhat comparable to the driver scheduling problem. Section 3.3 describes different types of optimization techniques, both exact methods and heuristics. The chapter ends with Section 3.4 describing the conclusions.

3.1. Scheduling and rostering

This section gives an introduction to scheduling and rostering and known problems regarding scheduling and rostering. The purpose of this section is to become familiar with several planning and scheduling aspects. The section starts with Section 3.1.1 giving an introduction. Section 3.1.2 elaborates on problems in scheduling personnel. Sections 3.1.3 and 3.1.4 describe staff scheduling and rostering and shift assignment, respectively. Sections 3.1.5 and 3.1.6 describe crew scheduling and crew rostering, respectively. Tour scheduling and the tour and shift labor scheduling problems are the topics in Sections 3.1.7 and 3.1.8.

3.1.1. Introduction

Rostering is primarily concerned with the allocation of jobs among a given workforce (Ernst et al., 2004). Fair and reasonable rostering plays a very important role in arousing worker’s enthusiasm and setting work productivity. Besides of that, it also brings great economic benefit (Zhang et al., 2007). The economic benefit incur the right staff level, time per shift and so on. In determining schedules and rosters for employees, there exist different stages. These stages are: demand modelling, days off scheduling, shift scheduling, line of work construction, task assignment, and staff assignment (Ernst et al., 2004).

3.1.2. Rostering or scheduling personnel problems

Creating a good roster for each employee becomes more and more complicated. These rostering problems are studied in the literature. Broadly speaking, these problems aim at determining the work schedule of each available employee over a planning horizon (Er-Rbib et al., 2020).

Several rostering problems aim at determining the staff level for each shift. In that case, the rostering problem corresponds to a generalized set covering problem (a given number of employees is required for each shift). Other problems that aim at one employee per duty, correspond to a set partitioning problem (Er-Rbib et al., 2020). An example is public transport, where one driver per bus is needed.

Set partitioning problems are well known to be computationally challenging for traditional single processor computing. One approach to improving tractability is to divide the problem into smaller sub problems that can be solved using multiple processors (Joseph, 2002). Another option can be parallel computing. In parallel computing, multiple processing elements are used to execute the program instructions simultaneously (Arkin et al., 2016). It is an effective method to improve the computing speed and processing power (Li et al., 2017). The set partitioning problem is known to be NP-hard, however, it is often used to model important real-world decision problems. Boschetti et al. (2008) state that the set partitioning problem formulation can be used to model many important real-life transportation problems. Scheduling of airline crews, bus crews, railway crews can be formulated as a set partitioning problem. Also, the scheduling of vehicles, ships and airline fleets can be formulated likewise (Boschetti et al., 2008).

3.1.3. Staff scheduling and rostering

A distinction can be made between cyclic and acyclic rostering. In a cyclic roster, all employees of the same class perform the same line of work, but with different starting times for the first shift or duty. In an acyclic roster the lines of work are independent due to demand fluctuations. This results in shifts with different lengths and starting times (Ernst et al., 2004).

3.1.4. Shift assignment

Shift assignment is a special case of tour scheduling. Tour scheduling is the process of choosing off days for workers and allocate shift in the working days to workers. However, shift scheduling is the case where the days off are given as inputs (Ernst et al., 2004). The purpose is usually to minimize or maximize objectives and satisfy constraints that arise from management, labour unions and employee preferences (Xue et al., 2018). The shift assignment problem is usually highly constrained and difficult to solve. The problem can become more complex if workers have mixed skills, if the start/end times of shifts are flexible and if multiple criteria are used for evaluating the quality of the solution.

3.1.5. Crew scheduling

Crew scheduling involves the selection of a best set of duties and is typically applied in transportation systems (Ernst et al., 2004). Bach et al. (2016) formulate the crew scheduling problem as a set covering problem and solve it using column generation. However Ezzinbi et al. (2014) propose to decompose the crew scheduling problem into 2 parts, the crew pairing and crew assignment (rostering). The crew pairing problem is formulated as a set partitioning problem. The crew assignment is approached by using the basic model given in Gamache & Soumis (1998). This basic model looks as follows:

$$Min \sum_{k \in K} \sum_{s \in S^k} c_s^k x_s^k$$

Subject to

$$\sum_{k \in K} \sum_{s \in S^k} \gamma_p^s x_s^k \geq n_p \qquad \forall\, p \in P$$

$$\sum_{s \in S^k} x_s^k = 1 \qquad \forall\, k \in K$$

$$x_s^k \in \{0, 1\}$$

x_s^k: $is\ 1\ if\ crew\ member\ k\ is\ assigned\ to\ schedule\ k\ and\ 0\ otherwise$

K: $the\ set\ of\ crew\ members\ with\ a\ certain\ skill$

S^k: $the\ set\ of\ work\ schedules\ that\ are\ feasible\ for\ employee\ k\ \in K$

P: $the\ set\ of\ skills$

n_p: $represents\ the\ minimum\ number\ of\ crew\ members\ that\ must\ be\ assigned\ to\ skill\ p\ \in P$

γ_p^s: $is\ 1\ if\ skill\ p\ \in P\ is\ included\ in\ schedule\ s\ and\ 0\ otherwise$

c_s^k: $the\ cost\ of\ schedule\ s \in S^k\ for\ employee\ k \in K\ represents\ the\ schedule\ cost$

The main purpose for the crew scheduling is to minimize the total cost. The total cost is calculated by summing the different crew assignment cost.

3.1.6. Crew rostering

Crew rostering is a process to generate a timetable for crew members that is aligned with certain guidelines regarding health and safety policies. A roster is created by assigning crew pairing to individual crew members with respect to constraints and regulations such as the maximum working hours (Limlawan et al., 2011). Xie et al. (2017) describe crew rostering in public bus transit as constructing personalized monthly schedules for all drivers. The problem is formulated as a multi-objective problem that takes both the company and driver interests into account. The paper solves the problem using ant colony optimization, simulated annealing and tabu search methods. Peng et al. (2016) solve the crew rostering in two stages. The first stage is the creation of rosters per week without involving drivers. The second stage is the assignment of drivers to the created rosters.

3.1.7. Tour scheduling

Tour scheduling is the process of choosing off days for workers and allocate shifts in the working days. Pan et al. (2018) uses mixed-integer linear programming to model the tour scheduling problem. To find good quality solutions they propose to use a hybrid heuristic, which combines tabu search and large neighbourhood search techniques. Rong (2010) compares two models for the tour scheduling problem. The approach of using a general integer programming formulation is compared with a binary integer programming formulation. The paper also takes mixed skills of workers into account. Ni & Abeledo (2007) formulate the tour scheduling problem as a set covering problem. The paper solves the problem by decomposition and a branch-and-price approach. Also Kheiri et al. (2021) use a branch-and-price approach to solve large-scale employee tour scheduling problems.

3.1.8. Tour and shift labor scheduling problem

The tour and shift labor scheduling problem focuses on seeking a minimum number of employees that correspond with the assigned shift schedules of the employees. This is done to satisfy fluctuating demand requirements. Morris & Showalter (1983) give the following formulation for the tour and shift labor scheduling problem:

$$Min\ Z = \sum_{t \in T} x_t$$

Subject to

$$\sum_{t \in T} a_{tp} x_t \geq r_p \qquad for\ p\ \in P$$

$$x_t\ \geq 0\ \ and\ integer\ for\ t \in T$$

The x_t represents of employees at time t. a_{tp} is a binary variable that indicates if t is a working period in tour p and r_p represent the required staffing level of working period p. The objective of the tour and shift labor scheduling is cost-driven. It is a very simple version of the problem since it only has to comply with the constraint that sufficient employees are present in all periods.

3.2. Comparable problems

In literature, specific problems are studied more in detail. This section handles some of these specific problems that are comparable with our problem. Section 3.2.1 and 3.2.2 describe the nurse scheduling problem and the airline crew scheduling problem, respectively. Section 3.3.3 elaborates on the bus driver rostering problem. Section 3.2.4 describes the set partitioning problem.

3.2.1. Nurse Scheduling problem

The nurse scheduling problem, also known as the Nurse Rostering Problem is a combinatorial optimization problem that lies in constructing an optimal set of shift assignments (Farasat & Nikolaev, 2016). The result is a schedule that satisfies constraints while being seen to be fair by the staff concerned (Aickelin & Dowsland, 2004). The nurse scheduling problem is shown to be NP-hard in general (Osogami & Imai, 2000).

Nurse scheduling is the assignment of shifts to nurses over several days. This is a very time-consuming job and usually executed by the head nurse (Legrain et al., 2014). However, specialties of nurses and sharing resources is adding extra complexity to the planning. Due to this, the need for automatic or computer-aided planning methods increases. Also Leung et al. (2021) states that the increasing need for better workload distribution has made nurse scheduling critical. However, usually nowadays they still rely on human experience, often leading to ineffective planning.

Kheiri et al. (2021) formulate the nurse rostering problem as an integer program which takes preferences into account. When a preference is not satisfied, this is counted as a violation. Different type of objectives can be pursued, for example minimizing the extra expenses that the hospital must pay for hiring extra nurses (J. Lim et al., 2012). Other objectives can be minimizing the cost or number of violations of preferences.

Both Dowsland (1998) and Oughalime et al. (2008) propose a tabu search for solving the nurse scheduling problem. Dowsland applies Tabu search due to its robustness and the ease with which it can be adapted to embrace minor changes in the problem. Oughalime uses tabu search since the method has proven to be very effective on a variety of problems.

Knust & Xie (2017) use simulated annealing to solve the nurse rostering problem. Simulated annealing is chosen here, since the method has proven itself to be robust and it is fairly easy to implement. Hadwan & Ayob (2010) show promising results when solving the nurse rostering problem with the help of simulated annealing.

The nurse scheduling problem has several common aspects when we compare it with our problem. Our problem can be categorized as a combinatorial optimization problem which focuses on an optimal set of shift assignments. The sharing resources in the nurse scheduling problem is similar with the trucks in our problem. Furthermore, the minimization of extra expenses, workload distribution and taking into account preferences are also aspects that match.

3.2.2. Airline crew scheduling and rostering

Scheduling and rostering is also in the airline industry a challenge. Airline crew scheduling problems in literature are usually approached to obtain good solutions in reasonable time instead of solving to optimality (Cappanera & Gallo, 2004). This is also the reason most approaches are based on heuristics. The heuristics approach is used since the airline crew assignments are day-to-day activities and thus obtaining solutions quick is more important than obtaining the optimal solution.

Maenhout & Vanhoucke (2010) visualize the airline crew scheduling problem as a network structure for a certain time horizon. Figure 20 shows this visualization. The ‘s’ on the left hand side is the start point and the ‘t’ on the right hand side is the end point. The black boxes represent the working days for the employees. The ‘f’ in the bottom of the network represent a free day for the worker. Each flow line from the start point to the end point gives a roster for a certain time period, in this case 4 days.

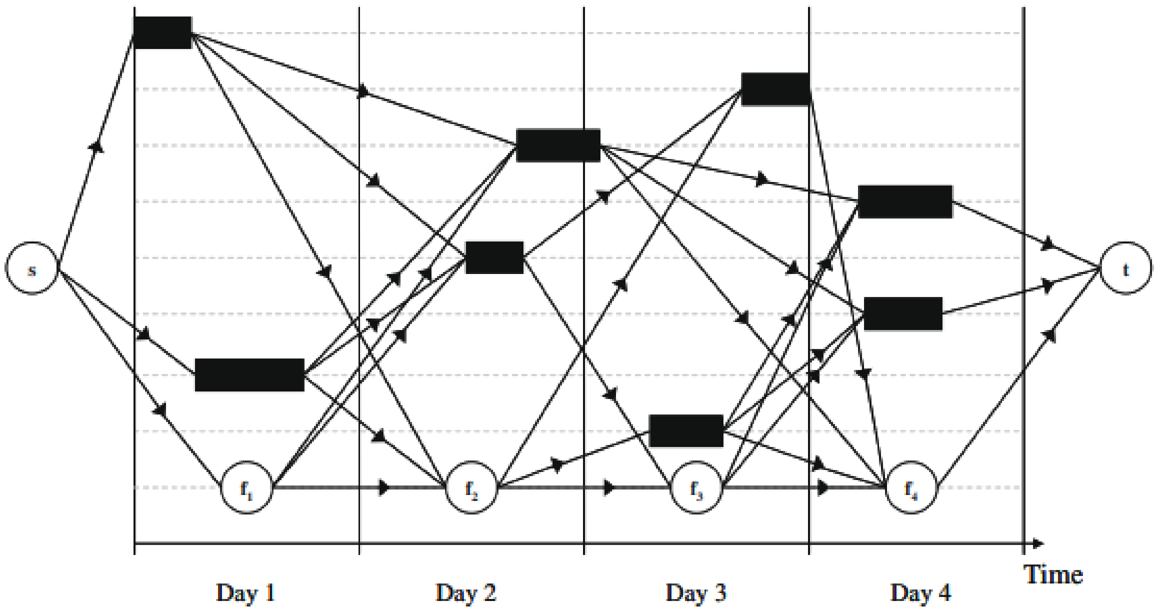

Figure 20: Visualization airline crew scheduling problem (Maenhout & Vanhoucke, 2010)

The first aspect in the airline crew rostering problem is that obtaining good solutions quick is important. Our problem lies between the tactical and operational levels, so it can be more important to find good feasible solutions quick rather than seeking for optimality. Also the buildup of a roster for drivers/airline crew can be applied to our problem.

3.2.3. Bus driver rostering problem

In the public bus transport industry, it is estimated that the cost of a driver schedule accounts for approximately 60% of a transport company's operational expenses (Perumal et al., 2019). Rostering drivers deals with the assignment of duties to workers along a planning horizon of a specified length, usually four or more weeks (Respício et al., 2013).

Respício et al. (2013) also propose a multi-objective approach for the Bus Driver Rostering Problem (BRP) that takes into account both the interests of the driver and the company. This is done since it is undeniable that the conflicting interests of both employer and employee must be considered. The employer's interests are usually related to costs and the employee can have preferences regarding his roster that influence the cost. Barbosa et al. (2013) propose a hybrid approach of column generation and genetic algorithms to achieve good quality rosters in short time. This is done since the BRP is classified as a NP-hard problem. Barbosa et al. (2015) propose a decomposition model implemented in a framework and then the usage of column generation to optimize.

Er-Rbib et al. (2020) use the following feasibility rules to create a roster for a bus driver schedule:

1. Depending on the roster type, there must be exactly 2 or 3 days off per employee, with at least two consecutive days off in each position.
2. The number of consecutive days without a day off cannot exceed 6.
3. There must be at least 9 hours of rest between two consecutive duties.
4. There must be at least 10 hours of rest between two consecutive duties if they are followed by a third shift.
5. A 57-hour period of rest, including at least two consecutive days off, must be assigned to each position.
6. In any period of 28 consecutive days ending with a working day, the average rest time between the duties must be at least 12 hours.

7. At most one duty lasting 13 hours or more can be assigned to each position.

These feasibility rules result in corresponding constraints when defining a mathematical model. The rules can differ between industries, companies and countries. However, the essence of the feasibility rules is usually the same.

The bus driver rostering problem has the most common aspects with our problem. The first common aspect is the industry where the problem exists. The operational expenses play a big role in the need for a good schedule for drivers. Also the interest of both the driver and the company that have to be taken into account is a common aspect in the bus driver rostering problem and our problem. The feasibility rules are comparable with the restrictions existing in our problem.

3.2.4. Set partitioning problem

Set partitioning problems occur as subproblems in various combinatorial optimization problems (Müller, 1998). A set partitioning problem determines how the items in one set (S) can be partitioned into smaller subsets. The complete set partitioning problem (CSP) is the zero-one integer program (Lin & Salkin, 1983):

$$z = \sum_{j \in J} c_j * x_j$$

Subject to:

$$\sum_{j \in J} a_j * x_j = 1 \;\; \forall r \; \in R$$

$$x \; \in \{0,1\} \, \forall j \in J$$

Where a_j is the m column (a_{ij}) of zeros and ones. c_j is the cost and always ≥0. x_j is the binary decision variable which indicates if the corporation is used ($x_j = 1$) or not ($x_j = 0$). The objective is to find minimal cost partition of the root set (Krieken, 2006).

Several problems are solved using set partitioning formulations. The set partitioning problem formulation is used to solve Air scheduling problems (Rushmeier et al., 1995), team formulation problems (Daş et al., 2021) and vehicle routing problems (Friedrich & Elbert, 2022).

Our problem has common aspects with the set partitioning problems. The objective of our problem is minimizing the cost of a schedule by assigning the drivers to the right shifts. The set partitioning problem is NP-hard. However, in our problem not all shifts need to be executed by company drivers, since a part of the shifts are executed by temporary workers or external transporters. This results in an easier problem that possibly is not NP-hard.

3.3. Optimization Methods

This section handles several optimization methods to solve different rostering problems. The section considers both exact methods and heuristics. The section handles the exact methods; enumeration, mathematical programming, branch and bound, column generation and brand and price in Sections 3.3.1 to 3.3.5. Sections 3.3.6 to 3.3.8 describe the heuristics: constructive heuristic, simple local search and meta-heuristics.

3.3.1. Enumeration

Enumeration is a simple method in which a complete or partial enumeration of all possible solutions is carried out (Ernst et al., 2004). However, complete or even partial enumeration is with larger instances not executable anymore in reasonable time.

3.3.2. Mathematical programming

Mathematical programming is the branch of mathematics dealing with techniques for maximizing or minimizing an objective function subject to linear, non-linear and integer constraints on the variables (Dantzig, 1986). In mathematical programming the models seek to minimize or maximize an objective that is subject to a set of constraints (Ernst et al., 2004). There exist several types of mathematical programming, including linear programming and integer programming. Many network flow problems are solved by the use of mathematical programming. Er-Rbib et al. (2020) use mathematical programming in their problem formulation of the bus driver rostering problem to generate input for a commercial solver to find a solution. Hasebe et al. (2017) use mathematical programming to generate "good" solutions for the nurse scheduling problem. A good solution is defined as a solution that easily can be modified to a subjectively ideal schedule.

3.3.3. Branch and Bound

A Branch and Bound algorithm searches the complete space of solutions for a given problem for the best solution (Clausen, 1999). The use of bounds for the function enables the algorithm to search parts of the solution space implicitly, since complete enumeration is usually impossible. Dawid et al. (2001) use Branch and Bound with a variable branching strategy to solve airline crew rostering problems, which show promising results.

3.3.4. Column generation

Column generation is a computational technique for solving large-scale integer linear programming problems or linear programming problems. Column generation uses linear programming relaxations and aims to reduce the computational effort of exploring complete branch and bound trees (Nishi et al., 2011).

3.3.5. Branch and Price

Branch and Price methods are used in several papers for solving rostering problems. Branch and Price is a combination of the Branch and Bound and the Column Generation methods. The algorithm applies a column generation algorithm in every node of the branching tree to find the optimal LP solution and a branching method to drive the LP solution, when fractional, to integrality (Akbarzadeh & Maenhout, 2021). Horváth & Kis (2017) use Branch and Price to solve the crew scheduling problem to obtain feasible integer solutions. Also Freling et al., 2004 use the Branch and Price algorithm and they show the algorithm is obtaining better solutions than using only Branch and Bound.

3.3.6. Constructive heuristics

When it is more important to get a sensible feasible solution quickly than to invest a lot of time into finding the optimal solution, constructive heuristics are used. The solution generated by constructive heuristics can either be used as a quick solution or as a starting point for obtaining a better solution. To obtain better solutions out of the starting solution, local search algorithms are used.

3.3.7. Simple local search

A simple local search uses an initial solution as input and tries to improve. Local search is basically a single-objective optimization technique for finding a single optimal solution (Ishibuchi et al., 2008). The improvements happen iteratively by exploring feasible solutions in the neighbourhood of the current solution. Hill-climbing and descent are two examples of simple local search (Ernst et al., 2004).

However, simple local searches do not use complicated moving strategies employed in many meta-heuristics.

3.3.8. Meta-heuristics

Glover & Laguna (1997) describe a meta-heuristic as a local heuristic search procedure that explores the solution space beyond local optimality. Two very known meta-heuristics are simulated annealing and tabu search. The idea of simulated annealing comes from the energy minimizing process of the cooling of metals. At high temperatures the algorithms accepts worse solutions with a higher probability than at lower temperatures. This probability converges to zero when the temperature decreases (Ernst et al., 2004). Lučic & Teodorovic (1999) use simulated annealing to solve their multi-objective crew rostering problem, which shows improvements on the initial solutions.

Tabu search is a meta-heuristic that incorporates adaptive memory and responsive exploration (Glover & Laguna, 1997). Tabu search uses the adaptive memory to keep track of a list of forbidden movements, the tabu list. The tabu list prevents the algorithm from cycling between known solutions. Ikegami & Niwa (2003) first solve the problem by using branch and bound, but later choose tabu search as an efficient heuristic algorithm to solve the nurse scheduling problem.

3.4. Conclusions

This chapter answered the sub-question: *"What theory and methods exist in literature to improve scheduling personnel?"*

Scheduling problems are well-known in the literature. Different solution methods are focused on solving these problems. We saw different rostering and scheduling problems, including staff assignment, crew scheduling and tour schedule. These problems are mostly modelled using (mixed-) linear programming as a set covering or set partitioning problem, depending on the objective. These problems are usually solved by simulated annealing, tabu search or column generation.

We also analysed specific types in scheduling problems in different industries: nurse scheduling, airline crew rostering, bus driver scheduling and the set partitioning problem. The solutions to these problems are focused on obtaining good solutions in a reasonable time instead of obtaining the optimal solution. This is since scheduling is a daily task and thus methods with a long calculation time are not desirable. The long calculation time for finding the optimal solution is caused by that these problems are usually NP-hard problems. For this reason, heuristic approaches are used, such as simulated annealing and tabu search.

Finally, we introduced methods for optimizing. The chapter contains both exact methods and heuristics. The heuristics can be constructive heuristics or heuristics that optimize an initial solution by iterative improvements. Heuristics are mostly used to find good solutions quickly while exact methods are used to find the best possible solution.

4. Solution design

This chapter answers research question: "How to build a weekly driver schedule for the company with the aim of lowest cost possible?"

The chapter starts with the company situation in Section 4.1. Section 4.2 describes the model with its aspects included. In Section 4.3 we present the mathematical formulation of the model. We describe the input and output of the model in Section 4.4. Section 4.5 describes if preprocessing a specific constraint is beneficial. The chapter finalizes with an illustrative example given in Section 4.6 and conclusions in Section 4.7.

4.1. Company situation

The situation at the company has some similar characteristics with the situations described in the literature. The feasibility constraints of Er-Rbib et al. (2020) have common grounds with our problem. These constraints arise from legal requirements or from driver preferences and can be modelled as mathematical constraints. The different start times in an acyclic roster described by Ernst et al. (2004) are also applicable to our problem, where each shift can have different start and/or end times. Our problem has common aspects with a set partitioning problem where one employee per duty is the case. However, since a part of the shifts in our problem is executed by temporary workers or external transporters, not all shifts have to be filled in by company drivers. To the best of our knowledge, that is something that we could not find in literature and thus lacks.

4.2. Model description

We want to develop a method that generates a roster for each driver that satisfies the restrictions of the driver and has a minimal deviation between scheduled and contractual hours. The model has to be used along with the new advanced planning system. This means we use the output of the advanced planning system as input for our model.

So, the objective of the model is to minimize the deviation between scheduled and contractual hours while satisfying restrictions. The scheduling restrictions are already mentioned in Section 2.2.2. The following restrictions are incorporated in the model:

Working day: The drivers already have a roster indicating which days to work in a specific week. This means that each driver needs to have a shift assigned on each working day.

Skill level: The skills indicate the ability to execute fresh shifts. As explained in Chapter 2, we have 3 types of shifts: fresh, non-fresh and mixed. We categorize the skills in 2 levels. If a driver has skill level 1, the driver can only execute non-fresh shifts. A driver with skill level 2 can execute non-fresh, fresh and mixed shifts.

Start time: A number of drivers have agreements regarding the start time of the shifts. This means that the shifts assigned to the driver need to start after the minimum and before the maximum start time.

End time: The restriction regarding the end time indicates that a shift should end before the agreed maximum end time of a driver.

Total time: The shift should not be longer than the agreed maximum total shift duration of a driver.

Maximum workload: A number of drivers have agreed on a maximum workload in a shift due to medical conditions. This should be ensured in the restriction regarding the maximum workload. To add up on that, the drivers also have a restriction regarding the maximum average workload in a week.

Deviation in start time: The restriction regarding the deviation in start times ensures that there is no more than 2 hours of deviation in start time between 2 consecutive working days.

4.3. Mathematical model

In this section we present the mathematical model. We start with giving the sets, parameters and the decision variables in Section 4.3.1. Section 4.3.2 covers the objective function. Section 4.3.3 describes the constraints of the model, where we use pre-processing techniques to reduce the problem size. Finally, in Section 4.3.4 we discuss how we solve the model.

4.3.1. Sets, parameters and decision variables

Sets

$I = set\ of\ drivers\ (i\ \in I)$
$J = set\ of\ shifts\ (j\ \in J)$
$D = set\ of\ days\ (d\ \in D)$

Parameters

$MinStart_i = Minimal\ start\ time\ of\ driver\ i$
$MaxStart_i = Maximal\ start\ time\ of\ driver\ i$
$MaxEnd_i = Maximal\ end\ time\ of\ driver\ i$
$Contract_i = Contractual\ hours\ of\ driver\ i$
$MaxTrip_i = Maximum\ numbers\ of\ trip\ for\ driver\ i$
$WorkingDay_{id} = 1\ if\ driver\ i\ is\ working\ on\ day\ d, 0\ otherwise$
$SkillLevel_i = SkillLevel\ of\ driver\ i$
$MaxAverageWorkload_i = Maximum\ average\ number\ of\ trips\ per\ week\ of\ driver\ i$
$MaxTime_i = Maximal\ shift\ time\ of\ driver\ i$

$StartTime_j = Starting\ time\ of\ shift\ j$
$EndTime_j = Ending\ time\ of\ shift\ j$
$TotalTime_j = Total\ time\ of\ shift\ j$
$Trips_j = Number\ of\ trips\ in\ shift\ j$
$NeededSkillLevel_j = Minimum\ Skill\ Level\ needed\ for\ job\ j$
$ShiftDay_{jd} = 1\ if\ shift\ j\ is\ on\ day\ d, 0\ otherwise$

Decision variables

$X_{ij} = 1\ if\ driver\ i\ is\ working\ shift\ j, 0\ otherwise$
$Undertime_i = Total\ undertime\ of\ driver\ i$
$Overtime_i = Total\ overtime\ of\ driver\ i$

4.3.2. Objective function

The objective of our model is to minimize the absolute deviation between scheduled and contractual hours. We formulate this by minimizing the undertime and overtime of the drivers:

$$Min\ Z = \sum_{i \in I} (Undertime_i + Overtime_i) \qquad 1$$

4.3.3. Constraints

This section explains the constraints in our model. We start with explaining our preprocessing phase where we generate input for our model by preprocessing our decision variables.

Preprocessing

In order to reduce the problem size, we apply preprocessing. An underlying idea of preprocessing is to analyze inequalities and establishing whether the inequality is redundant or whether the inequality forces some of the binary variables to either zero or one (Savelsbergh, 1994). By applying preprocessing, these decisions are made in advance. So, preprocessing reduces the problem size by providing input for the model.

Our input for preprocessing consists of already assigning values to decision variables that we already know. For example: if a driver i is not able to execute shift j due to the lack of the skill level, we already assign the value '0' to X_{ij}.

We apply preprocessing to the constraints regarding the skill level, maximum workload, start time, end time and total shift time:

Skill level

Each driver needs to have the minimal skill needed for a shift to execute the shift. The restriction on the skill level is preprocessed and we provide a '0' for the decision variable if the driver does not meet the skill level for a certain shift.

Maximum workload

A number of drivers have agreed to have a maximum workload per shift. The maximum workload, which is expressed in the number of trips per shift, is preprocessed. If a shift has a higher workload than the maximum workload of a driver, this shift cannot be assigned to that driver.

Minimal start time

We also preprocess the restriction on the minimal start time. If a shift starts earlier than the agreed minimal start time of a driver, we assign a value of '0' to this decision variable.

Maximum start time

We need to ensure that a driver gets a shift that starts before his or her maximum start time. We preprocess this restriction by providing a '0' as input for the decision variable when a shift starts later than the maximum start time of a driver.

Maximum end time

If a shift ends later than the maximum end time of a driver, this shift cannot be assigned to that specific driver. So, we assign a value of '0' to that decision variable.

Maximum total time

We need to ensure that a driver does not get a shift assigned that is longer than the maximum shift time of the driver. This is also done by preprocessing the decision variable.

Modelling formulation

It is possible that same combination of i and j result in that X_{ij} should be 0, resulting in redundancy. So, to avoid redundancy in the constraints, we group all constraints we want to preprocess. All constraints discussed above are preprocessed as follows:

$$\begin{gathered} If \\ NeededSkillLevel_j > SkillLevel_i \\ OR \\ Trips_j > MaxTrip_i \\ OR \\ StartTime_j < MinStart_i \\ OR \\ StartTime_j > MaxStart_{i_i} \\ OR \\ EndTime_{j_j} > MaxEnd_i \\ OR \\ TotalTime_j > MaxTime_i \\ Then \\ X_{ij} = 0 \;\; \forall\, i,j \end{gathered} \tag{2}$$

Calculation scheduled hours

In order to calculate the deviation between scheduled and contractual hours, we calculate the undertime and overtime in Constraint 3.

$$Undertime_i - Overtime_i + \sum_{j \in J} (TotalTime_j * X_{ij}) = Contract_i \; \forall \, i \tag{3}$$

Working day of a driver

Constraint 4 ensures that if a specific day is a working day for a driver, we assign a shift to this driver.

$$\sum_{j \in J} (X_{ij} * ShiftDay_{jd}) = WorkingDay_{id} \; \forall \, i, d \tag{4}$$

At most one driver per shift

Constraint 5 avoids that multiple drivers are assigned to 1 shift.

$$\sum_{i \in I} X_{ij} \leq 1 \; \forall \, j \tag{5}$$

Maximum average workload

Some drivers have agreed to have a maximum average workload per week. Constraint 6 ensures that the average workload does not exceed the maximum average.

$$\sum_{j \in J} (Trips_j * X_{ij}) \leq \sum_{d \in D} WorkingDay_{id} * MaxAverageWorkload_i \; \forall \, i \qquad 6$$

Maximum 2 hours of deviation in start times for 2 consecutive days

The start times between 2 consecutive days cannot deviate more than 2 hours. Constraint 7 shows how we model this.

$$If$$

$$WorkingDay_{id} + WorkingDay_{id+1} = 2 \qquad 7$$

$$Then$$

$$\sum_{j \in J} (X_{ij} * StartTime_j * ShiftDay_{jd}) - \sum_{j \in J} (X_{ij} * StartTime_j * ShiftDay_{jd+1}) \leq 2{:}00 \; \forall \, i, d = 1..5$$

$$\sum_{j \in J} (X_{ij} * StartTime_j * ShiftDay_{jd}) - \sum_{j \in J} (X_{ij} * StartTime_j * ShiftDay_{jd+1}) \geq -2{:}00 \; \forall \, i, d = 1..5$$

It is also possible to preprocess the constraints regarding the deviation in start times. In the preprocessing phase we create pairs with shifts that deviate more than 2 hours and thus cannot be assigned to one driver. However, this results in more constraints. In Section 4.6 we test if preprocessing the deviation in start times results in a reduction of running time.

Binary variable

Constraint 8 indicates the binary property of our decision variable.

$$X_{ij} \in \{0,1\} \qquad 8$$

Non-negativity

All other variables are non-negative, we ensure this in Constraint 9.

$$All \; other \; variables \; \geq 0 \qquad 9$$

4.3.4. Solving the model

To solve the mathematical model, we use a solver. The main advantage of a solver is that it produces the best possible solution to a problem. So in terms of finding optimal solutions, it outperforms heuristics. Another advantage of using mathematical modelling in combination with a solver is the ease of changing goals or constraints. However, optimization with a solver can result in long running times when the problem becomes too complex (NP hard).

4.4. Model input and output

Input

For the model we have 2 'types' of input. The first type of input is the list of shifts that have to be executed in a specific week. Each shift has the following properties: start time, end time, total time, workload, day of the shift and type of shift. This list is an output of the advanced planning system which will be implemented in the future. This list is used as input since the model needs to be used along with the advanced planning system.

The second type of input is information regarding the drivers. This includes the working days of a driver and the restrictions of the drivers: start time, end time, skills, maximum workload, etc.

Output

The output of the model is a roster for each driver that indicate which shifts he or she has to work. This roster contains the workload, type of shift, start times, end times and scheduled hours. It is also possible to show what stores the driver has to visit. However, the content of the shift is less relevant for our research.

4.5. Illustrative example

We use week 2 of 2021 as illustrative example. This week contains a total of 564 shifts and 71 drivers who work a total of 240 shifts. We solve the scenario to optimality. This results in values for X_{ij}, indicating if driver i has to work shift j ($X_{ij} = 1$), or not ($X_{ij} = 0$).

We take one driver as an example. This driver has the following restrictions (Table 2):

Category	**Value**
Contractual hours (including breaks)	44
Minimal start time	07:00
Maximum start time	09:00
Maximum workload	No restriction
Skills	2
Maximum average workload	2.5
Maximum end time	No restriction

Table 2: Restrictions example driver

Figure 21 shows the output of the model. The figure shows that our example driver, driver 1 in this case, has to work shift 91, 183, 369 and 389.

Shift / Driver	1	2	3	90	91	92	182	183	184	368	369	370	388	389	390
1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0

Figure 21: Binary table indicating driver schedule

Table 3 presents the properties of the shifts and indicates the roster for the week for our driver.

Day	Shift	Start time	End time	Total time	Type	Workload
Monday	Free day					
Tuesday	91	07:45:00	19:00:00	11:15:00	2	2
Wednesday	389	07:35:00	17:40:00	10:05:00	1	3
Thursday	Free day					
Friday	369	07:45:00	19:05:00	11:20:00	2	2
Saturday	183	07:45:00	19:05:00	11:20:00	2	2
Sunday	Free day					

Table 3: Schedule example driver

In Table 3, we see that the roster for the driver meets the constraints given in Table 2. The start time for all working days is between 7:00 and 9:00, the skill level of the driver is higher than needed and the average workload (2.25) does not exceed the maximum average workload of the driver (2.5). The total scheduled hours sum up to 44, which equals the contractual hours of the driver.

4.6. Preprocessing deviation in start times

As mentioned in Section 4.3.3, we test if preprocessing the restriction regarding the deviation in start times result in a model that solves faster. We perform the test on the same instance as the illustrative example is based on, week 2 of 2021.

If we compile the model without preprocessing the deviation in start times the model contains 16.731 constraints. Solving to optimality takes 284 seconds. Compiling the model with preprocessed deviation in start times gives us a model with 2.123.703 constraints and solves in 580 seconds to optimality. The big increase in constraints comes due to the fact that each pair of shifts that cannot be assigned to one driver, results in an extra constraint.

Due to the increase in running time, we decide to not preprocess the deviation in start times. This means that we execute all experiments in Chapter 5 with the deviation in start times modelled as in Constraint 7.

4.7. Conclusions

This chapter answered the following sub question: *"How to build a weekly driver schedule for the company with the aim of lowest cost possible?"*

To summarize, the literature lacks a sort of set partitioning problem where not all shifts have to be filled. We formulate a mathematical model that can build a weekly driver schedule. We reduce the problem size with the help of pre-processing techniques, where we provide input for the model. The restrictions of the drivers are incorporated into constraints, these restrictions are: working day, skill level, start time, end time, total time and workload. The aim of the lowest cost possible is covered by the fact that we aim to minimize the deviation between scheduled and contractual hours.

The model is solved using a MILP solver, which results in the best solution for the problem given the constraints. The output of the advanced planning system functions as the first type of input for the model. The second type of input for the model is an overview of the restrictions of the drivers. The output of the model is a roster for each driver that has a minimal deviation between scheduled and contractual hours.

5. Experimental design

This chapter answers the following research question: “How does the method perform (compared to the current situation)?”

The chapter starts with Section 5.1 describing the experimental setting. Section 5.2 explains the experiments we execute. We add extra constraints to the model in Section 5.4. The incorporation of paid waiting time is done in Section 5.5. Section 5.6 presents the model with soft constraints regarding the start times. We compare the performance of the models in Section 5.7 and Section 5.8 shows the comparison between a license-based solver and a free solver. Sections 5.9 and 5.10 present a complete analysis of 2020 and a sensitivity analysis, respectively. Section 5.11 gives an analysis about the content of the rosters. The chapter finalizes with conclusions in Section 5.12.

5.1. Experimental setting

In this section we describe settings when executing the experiments. We perform all experiments under the same conditions. Below we describe this experimental setting.

Hardware

The settings for all experiments are the same regarding hardware. We execute all calculations on an Intel Core i7-8750H processor.

Instances

For the experiments in this chapter, we use weeks 2 to 12 from 2021. These are the weeks that are used for the analysis in Chapter 2 and thus are ready to compare with our model. As mentioned in Chapter 2, Week 2 to 12 are ‘normal’ weeks, which means that no special days are in these weeks. These normal weeks give the best reflection of reality.

Table 4 shows some characteristics of the instances. The first column ‘Week’ indicates the week number. The second column ‘Drivers’ shows the number of drivers present in that week. The column ‘Total shifts to fill’ is a sum of the working days of all drivers and indicates how many shifts have to be filled by the company drivers. The fourth column ‘shifts’ shows the total number of shifts available. The fifth column, current deviation per driver, is a result of the analysis in Chapter 2. The last column is the current total deviation in that week (= average deviation per driver * drivers) and is used for comparison. A more detailed overview of the drivers and shifts can be found in Appendix B.

Week	Drivers	Total shifts to fill	Shifts	Current deviation per driver (in hours)	Total deviation (in hours)
2	71	240	564	2.70	191.7
3	68	235	570	2.33	158.4
4	64	216	570	2.62	167.7
5	70	230	570	2.05	143.5
6	69	224	570	3.00	207.0
7	68	216	570	1.95	132.6
8	64	221	570	2.27	145.3
9	68	213	570	2.67	181.6
10	62	215	570	2.25	139.5
11	63	196	570	2.02	127.3
12	58	195	588	2.83	164.1
Average	66	218	571	2.43	159.9

Table 4: Characteristics instances

Software and solver

To model and solve our problem, we use mathematical modelling, which is motivated in Section 4.3.4. Mathematical models can be solved by the use of a (commercial) solver. Solvers are generally based on Branch and Bound algorithms. The solver we use to model and solve the problem is IBM CPLEX. This solver is implemented in Python using the CPLEX Python APO as a package in Python. CPLEX is a paid commercial solver, but we use a student license to solve the problem.

Running time

Our method has to create a schedule on a tactical level. The advantage of solving on a tactical level is that longer running times are affordable. Based on this knowledge, we decide to use a maximum running time of 1 hour (3600 seconds).

Results

The objective of our model is to minimize the total absolute deviation between scheduled and contractual hours. In each table we present the reduction compared to the current situation. The values of the current situation can be found in the column 'Total deviation' in Table 4.

5.2. Experiments

This section describes the experiments we perform in this chapter.

Basic model

The basic model is as described in Chapter 4. The basic model satisfies all constraints that are needed to be satisfied. These constraints concern mainly legal requirements.

Tighter constrained

The basic model complies with mainly legal requirements. However, extra constraints regarding maximum overtime and deviation in start times are desirable. These extra constraints result in a schedule that is comparable with the schedules created by the transport planners. We add the following constraints to the basic model:

- Maximum overtime of 10% per driver.
- Maximum 1 hour of deviation in start time between consecutive working days.
- Maximum 2 hours of deviation in start time through the whole week.

The maximum overtime constraint is needed to not result in too much overtime. This is since the schedule still has to be executed. Too much overtime in the schedule can result in even more overtime when the schedule is executed. A maximum of 10% is a rule of thumb in the company. The constraints regarding the deviation in start time are more desirable for the drivers themselves. Less deviation in start time is desirable.

In this model we consider all constraints to be hard constraints. This means that violating a constraint is not possible. In the experiment regarding the tighter constrained model we add constraints one by one to see the impact of each constraint separately.

Paid waiting time

In our second model we allow drivers to have paid waiting time. Paid waiting time exists when a shift starts later than the maximum start time of a driver. So, for example if a driver has a maximum start time of 9:00, but the shift starts at 10:00, paid waiting time arises. When we allow paid waiting time, this means that the driver starts at 9:00 and waits until 10:00 to start his or her shift. This hour that the driver is not working is paid and counts towards the scheduled hours of the driver. The driver can either be starting at 9:00 and wait until 10:00 or the driver can start at 10:00 and gets paid between 9:00 and 10:00. Whether the paid waiting time is at home or is at the company, does not matter for our result.

Since allowing paid waiting time can let drivers start later than their maximum start time, it expands the solution space of the model. A larger solution space results in, logically, more possible solutions. We investigate this since our model with hard constraints can possibly result in infeasible solutions, because hard constraints cannot be violated. To add up on that, paid waiting time can be compared to undertime. This is since the paid waiting time and undertime are both not worked, but have to be paid. So, it does not matter if undertime is prior to the shift (paid waiting time) or not.

Soft constraints

Our last model is the model where we consider the constraints regarding the start times to be soft constraints. Since the restrictions on start times are rather preferences than hard requirements, it is possible to violate them. The model is comparable to the model with paid waiting time. However, in this case the 'penalty' for violating a constraint does not count towards the scheduled hours of the driver.

Both the constraints regarding the minimum and maximum start time can be violated. So, it is possible for the drivers that they are scheduled to start before their minimum start time or after their maximum start time. To overcome that all constraints regarding start times are violated, we induce a penalty when a constraint is violated.

Overview models

Table 5 gives an overview of which constraints are included in each model. We separate 3 models: Considering hard constraints, allowing paid waiting time and considering soft constraints regarding the start times. All models include all constraints from the basic model and the extra constraints from the tighter constrained model (maximum 10% overtime, maximum 1 hour of deviation in start time between 2 consecutive days and maximum 2 hours of deviation in start time over the whole week). The model with paid waiting time has the possibility to allow paid waiting time, we add a constraint for this. The model where we consider the constraints regarding the start times to be soft constraints has the possibility to violate these constraints.

Constraint	Hard constraints	Paid waiting	Soft constraints
Basic model constraints	X	X	X
Max overtime	X	X	X
1 hour of dev	X	X	X
2 hour dev whole week	X	X	X
Paid waiting time allowed		X	
Soft constraints regarding start times			X

Table 5: Overview content models

License-based vs free solver

For all experiments we execute, we use a license-based solver. However, there are also free solvers available to solve our models. We compare the license-based solver with a free solver to see if the investment in the license-based solver is worth the investment.

Sensitivity analysis

In order to evaluate the robustness of our created schedules, we perform a sensitivity analysis on the deviation between scheduled and contractual hours after execution of the schedule. We carry out a Monte Carlo simulation. Since the objective is to minimize the deviation between scheduled and contractual hours, we let the execution time of the shift be a stochastic variable.

With the new total shift time, we 'simulate' the execution of a shift. We calculate the total realized time per week and compare this to the contractual hours to see what the deviations between executed and contractual hours are after execution. In this way we see if the created schedules still have a minimal deviation between scheduled and contractual hours after realization.

Incorporation 46-hour contracts

The feeling exists that specific drivers have a larger influence on the deviation between scheduled and contractual hours than other drivers. This larger influence can be due to the restrictions or contractual hours of the drivers. So, we look into the undertime and overtime hours of each driver to see what drivers have the biggest influence and why.

5.3. Basic model

As mentioned in Section 5.2, the current model is as described in Chapter 4. This basic model contains the legal requirements a schedule needs to satisfy. Table 6 shows the results of the basic model.

Week	Total deviation (in hours)	GAP	Running time	Reduction (in %)
2	9.52	0%	284	95.0%
3	6.92	0%	237	95.6%
4	9.22	0%	555	94.5%
5	8.42	0%	411	94.1%
6	7.72	0%	227	96.3%
7	4.82	0%	234	96.4%
8	4.92	0%	188	96.6%
9	5.32	0%	518	97.1%
10	4.02	0%	326	97.1%
11	4.62	0%	249	96.4%
12	1.2	0%	51	99.3%
Average	6.06	0%	298	96.2%

Table 6: Results basic model

It is important to note that in the basic model there does not exist a feasible solution for week 12 given the constraints. With trial and error we find out that the constraints regarding the start times results in infeasibility. When we allow one driver to start later than his maximum start time (which also is the case in the current situation), we end up with a total deviation of 1.2. Since in the current situation the driver is allowed to start later, we allow this as well.

So, if we only take into account the legal requirements, we realize an average total deviation of 6.06 hours per week. This is a reduction of 96.2% compared to the current situation. The instances all solve to optimality with an average running time of 298 seconds, where the maximum is 51 and 555 seconds are the minimum and maximum, respectively.

5.4. Tighter constrained

The basic model satisfies all legal requirements. However, the rosters can be made that they are more likeable by the drivers. For example the constraint regarding deviation in start times. Currently we have a maximum of 2 hours of deviation in start time between consecutive days. However, less deviation in start time is desirable. Also, the basic model has no restrictions regarding the deviation in start times when days are not consecutive. Also less deviation in start times is desirable here. We add the following (tighter) constraints:

- Maximum overtime of 10% per driver.
- Maximum 1 hour of deviation in start time between consecutive working days.
- Maximum 2 hours of deviation in start time through the whole week.

The constraints are added one by one and in the last experiment, we include them all together. In this way, we see the effect of each constraint separately.

Maximum 10% overtime per driver

Week	Solution value (in hours)	GAP (in %)	Running time (in seconds)	Reduction (in %)
2	9.52	0%	255	95.0%
3	6.92	0%	466	95.6%
4	9.22	0%	552	94.5%
5	8.42	0%	700	94.1%
6	7.72	0%	156	96.3%
7	4.82	0%	243	96.4%
8	4.92	0%	204	96.6%
9	5.32	0%	219	97.1%
10	4.02	0%	147	97.1%
11	4.62	0%	120	96.4%
12	1.2	0%	63	99.3%
Average	6.06	0%	284	96.2%

Table 7: Results adding constraint for max 10% overtime

Table 7 shows the results of adding the constraint where each driver can have a maximum of 10% overtime. We see that the results of each week is the same as without the extra constraint. This means that the added constraint of maximum 10% overtime is not binding in this case. The running time increases in some cases and decreases in other cases, on average the running time decreases a little (from 298 to 284 seconds).

Maximum 1 hour of deviation in start time between consecutive days

Week	Solution value (in hours)	GAP (in %)	Running time (in seconds)	Reduction (in %)
2	10.72	0%	810	94.4%
3	7.92	0%	433	95.0%
4	10.62	0%	663	93.7%
5	8.62	0%	525	94.0%
6	8.62	0%	1743	95.8%
7	5.42	0%	266	95.9%
8	4.92	0%	527	96.6%
9	6.02	0%	764	96.7%
10	5.32	0%	1430	96.2%

11	5.12	0%	722	96.0%
12	1.2	0%	83	99.3%
Average	6.77	0%	724	95.8%

Table 8: Results adding constraint for max 1 hour deviation in start time

Table 8 shows the results when we add the constraint that 2 consecutive days can have a maximum of 1 hour deviation in start time. We see that the tighter constraint of maximum 1 hour of deviation in start time instead of 2 has an effect on the result. The solution value is worse than before. This seems logical since the new solution space is a subset of the original solution space. Also in general the model also takes more running time to find the optimal solution (724 seconds compared to 298 seconds earlier). We conclude that this constraint is affecting the solution value and results in a higher deviation between scheduled and contractual hours.

Maximum 2 hours of deviation in start time through the whole week

Week	**Solution value (in hours)**	**GAP (in %)**	**Running time (in seconds)**	**Reduction (in %)**
2	10.62	0%	1011	94.5%
3	7.42	0%	1318	95.3%
4	10.12	0.7%	3600	94.0%
5	9.12	0%	1408	93.6%
6	8.42	0%	847	95.9%
7	5.52	0%	729	95.8%
8	5.22	0%	830	96.4%
9	6.22	0%	947	96.6%
10	5.22	0%	2853	96.3%
11	4.62	0%	321	96.4%
12	1.2	0%	133	99.3%
Average	6.70	0%	1272	95.8%

Table 9: Results adding constraint for max 2 hour deviation in start time through the whole week

Table 9 shows the results when we add constraints indicating that we can have a maximum deviation in start time of 2 hours over the whole week. Since we add extra constraints for deviations in start times, our solution space becomes smaller. Our new solution space is now a subset of our original solution space. So, it is logical that the new solution values are equal to or worse than the solution values of the original problem. The running time also increases significantly (from 298 to 1272 seconds on average). We conclude that this constraint is thus binding and resulting in a higher deviation between scheduled and contractual hours.

All 3 extra constraints

Week	**Solution value (in hours)**	**GAP (in %)**	**Running time (in seconds)**	**Reduction (in %)**
2	11.92	0%	1742	93.8%
3	8.32	0%	3502	94.7%
4	11.32	0%	3231	93.2%
5	9.32	0%	2165	93.5%
6	9.12	0%	1830	95.6%
7	5.72	0%	1665	95.7%
8	5.62	0%	1884	96.1%
9	6.42	0%	1213	96.5%

10	5.62	0%	881	96.0%
11	5.22	0%	1725	95.9%
12	1.2	0%	292	99.3%
Average	7.25	0%	1830	95.5%

Table 10: Results adding all 3 constraints

Table 10 shows the results for the model with all 3 extra constraints. Since the new solution space is a subset of the original solution space, it is logical to see that the solution values are worse than our original model. Also, the running times are significantly higher than the basic model. In adding the extra constraints with maximum 1 hours of deviation between consecutive days and 2 hours over the whole week separately, we concluded a higher deviation between scheduled and contractual hours. When adding all constraints together in the model, we see an even higher deviation between scheduled and contractual hours. Also the running time is on average the highest compared to adding only one constraint to the basic model.

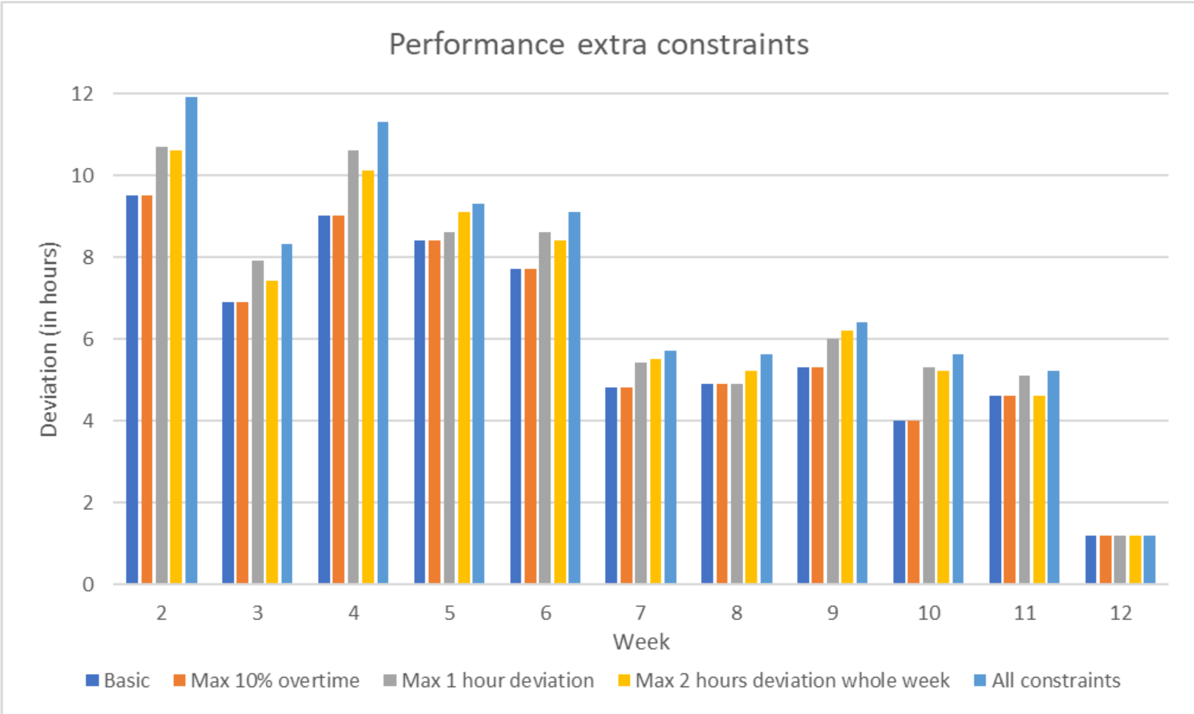

Figure 22: Comparison performance of the model considering extra constraints

Figure 22 shows us the comparison between the models visually. We see that the model with the tighter constraints performs the least best in all scenarios. However, despite the (expected) lower performance compared to the old model, our new model still realizes an average reduction of 95.5% in deviation between scheduled and contractual hours. The running time increases from 298 seconds to 1830 seconds. All experiments solve to optimality within the maximum running time of 3600 seconds. This model reflects real rosters more than our basic model given in Section 5.3. So, for the remainder of the experiments we include the extra (tighter) constraints.

5.5. Allowing paid waiting time

In this section we investigate if paid waiting time have a positive influence on feasibility and the outcome of the model. With including paid waiting time, we expand the solution space. The solution space is larger than the model with only hard constraints since all solutions generated in that model, are also feasible when paid waiting time are allowed. However, we now add the option that undertime prior to the shift is possible as well.

We use a weight of 1 for the paid waiting time. This means that 1 hour of paid waiting time has the same effect on the objective as 1 hour of overtime or undertime. This also means that we assume that 1 hour of paid waiting time is paid out as normal salary.

In order to implement the paid waiting time, we add the decision variable $PaidWaiting_{ij}$. This variable indicates the paid waiting time of driver i on shift j. We adjust the objective value function (2) and the constraints regarding the minimum and maximum start time.

Below we show the way we model the allowance of paid waiting time.

New objective value function:

$$Min\ Z = \sum_{i \in I} (Undertime_i + Overtime_i + \sum_{j \in J} (PaidWaiting_{ij})) \tag{10}$$

In our model from Chapter 4, we preprocess the maximum start time of a driver. If a shift starts later than the maximum start time, we assigned '0' to that decision variable. Now instead of assigning '0', we give the model the option to assign the shift. However, when the shift is assigned to that driver, the $PaidWaiting_{ij}$ receives a value. We model this as follows:

$$\begin{gathered} If \\ StartTime_j > MaxStart_{i_i} \\ Then \\ StartTime_j * X_{ij} \leq MaxStart_i + PaidWaiting_{ij} \forall\ i, j \end{gathered} \tag{11}$$

The paid waiting time counts towards the scheduled hours of the driver. Constraint 12 includes the paid waiting time in the scheduled hours:

$$Undertime_i - Overtime_i + \sum_{j \in J} (TotalTime_j * X_{ij} + PaidWaiting_{ij}) = Contract_i\ \forall\ i \tag{12}$$

It is important to note that this way of modelling only works if the paid waiting time has a higher or equal weight as undertime ($Weight\ PaidWaiting_{ij} \geq Weight\ Undertime_i$). So, if the weight for the paid waiting time is lower than the cost for undertime, the model does not work properly. This is due to the fact that with a lower weight for paid waiting time, the model will choose paid waiting time instead of undertime in the calculation for scheduled hours. However, luckily the paid waiting time are highly unlikely to have a lower weight than undertime, since it is undertime prior to the shift.

The table below (Table 11) shows the results of the instances when we allow paid waiting time.

Week	**Solution value**	**Paid waiting time (in hours)**	**GAP (in %)**	**Running time (in seconds)**	**Reduction (in %)**
2	10.52	0.3	8.8%	3600	94.5%
3	8.02	0.3	0%	3365	94.9%
4	10.62	1.6	8.3%	3600	93.7%

5	9.42	0.1	1.2%	3600	93.4%
6	9.12	0	1.1%	3600	95.6%
7	5.72	0	0%	1455	95.7%
8	5.52	0.1	0%	3292	96.2%
9	6.42	0	0%	1160	96.5%
10	5.62	0	0%	1140	96.0%
11	5.22	0	0%	1233	95.9%
12	1.2	0	0%	141	99.3%
Average	7.04	0.22	1.8%	2383	95.6%

Table 11: Results allowing paid waiting time

In the table we see that the average total deviation between scheduled and contractual hours is 7.04. From these 7.04 hours, we have an average of 0.22 hours of paid waiting. This is a reduction of 95.6% in deviation between scheduled and contractual hours. Since the paid waiting time variable has a value, this means that the model uses the variable to find better solutions after we expand the solution space by allowing paid waiting time.

An advantage is that there exist a feasible solution for every instance, without applying adjustments to the input and/or model. In contrast to the model with hard constraints, the model with paid waiting time does have a feasible solution for instance 12. Since the solution space is larger, we see that the model takes more time to solve the problem. The average running time increases from 1830 seconds to 2383 seconds. Also, some instances do not solve to optimality. 4 of the 11 instances cannot solve to optimality within the maximum running time of 3600 seconds, resulting in an average gap of 1.8% over all instances.

5.6. Soft constraints

Since the restrictions regarding the start times are rather preferences than hard restrictions, we can consider them as soft constraints. So, in this section we formulate the restrictions regarding the start times as soft constraints. This means that it is possible for the model to violate these constraints. However, violation of a constraint induces a penalty. After discussion with the operational transport planners, we decided to set the weight of the penalty to 1. This means that 1 hour of violation (either starting later or earlier than the constraint indicates) has the same influence on the model as 1 hour of overtime or undertime.

To model the soft constraints, we add $Penalty1_{ij}$ and $Penalty2_{ij}$ as variables. $Penalty1_{ij}$ receives a value if shift j is assigned to driver i and starts earlier than the minimal start time of this driver. $Penalty2_{ij}$ indicates the penalty if shift j is assigned to driver i and this shift starts later than the maximum start time of this driver.

An advantage is that with soft constraints the model can create more solutions than with hard constraints (solution space is larger since it is possible to violate constraints).

We add the penalty costs to the objective function and incorporate them into the constraints regarding the start times. We model this as follows:

New objective value function:

$$Min\ Z = \sum_{i \in I} (Undertime_i + Overtime_i + \sum_{j \in J} (Penalty1_{ij} + Penalty2_{ij})) \qquad 13$$

Now instead of assigning '0' to shifts that start earlier than the minimum start time for a driver, we give the model the option to assign the shift. However, when the shift is assigned to that driver, the $Penalty1_{ij}$ receives a value. We model this as follows:

$$
\begin{gathered}
If \\
StartTime_j < MinStart_i \\
Then \\
StartTime_j * X_{ij} \geq MinStart_i * X_{ij} - Penalty1_{ij} \forall\, i, j
\end{gathered}
\tag{14}
$$

The model also has the option to assign shifts to a driver that start later than the maximum start time of a driver. Now $Penalty2_{ij}$ is induced when the shift is assigned to the driver.

$$
\begin{gathered}
If \\
StartTime_j > MaxStart_i \\
Then \\
StartTime_j * X_{ij} \leq MaxStart_i + Penalty2_{ij} \forall\, i, j
\end{gathered}
\tag{15}
$$

Table 12 shows the results of the instances if we consider the constraints regarding the start times to be soft constraints.

Week	**Solution value**	**Deviation (in hours)**	**Penalty**	**GAP (in %)**	**Running time (in seconds)**	**Reduction (in %)**
2	10.32	9.92	0.4	6.8%	3600	94.8%
3	7.92	7.72	0.2	0.0%	1947	95.1%
4	9.42	8.02	1.4	1.1%	3601	95.2%
5	9.82	9.82	0	5.9%	3601	93.2%
6	9.12	9.02	0.1	0.0%	1718	95.6%
7	5.72	5.72	0	0.0%	910	95.7%
8	5.52	5.42	0.1	0.0%	2224	96.3%
9	6.42	6.42	0	0.0%	1501	96.5%
10	5.62	5.62	0	0.0%	2888	96.0%
11	5.22	5.22	0	0.0%	2779	95.9%
12	1.2	1.2	0	0.0%	158	99.3%
Average	6.93	6.74	0.2	1.3%	2266	95.8%

Table 12: Results considering soft constraints

When we consider the constraints regarding the start times to be soft constraints, we find an average deviation between scheduled and contractual hours of 6.74 hours. This is a reduction of 95.8% compared to the current situation. However, it is important to note that this (lower) total average deviation comes with an average penalty of 0.2. This means that on average 0.2 hours per week is violated regarding the start time preferences.

The model with soft constraints finds also feasible solutions for each scenario without making adjustments to the model and/or input. The running time is increasing compared to the model with hard constraints. The average running time is 2266 seconds instead of 1830 seconds for the model with hard constraints. Also, in 3 of the 11 instances the model do not solve to optimality within the maximum running time. This results in an average gap of 1.3% over all instances.

5.7. Comparison models

This section presents the comparison of the models described in Sections 5.4, 5.5 and 5.6. Figure 23 shows the results of the models visually.

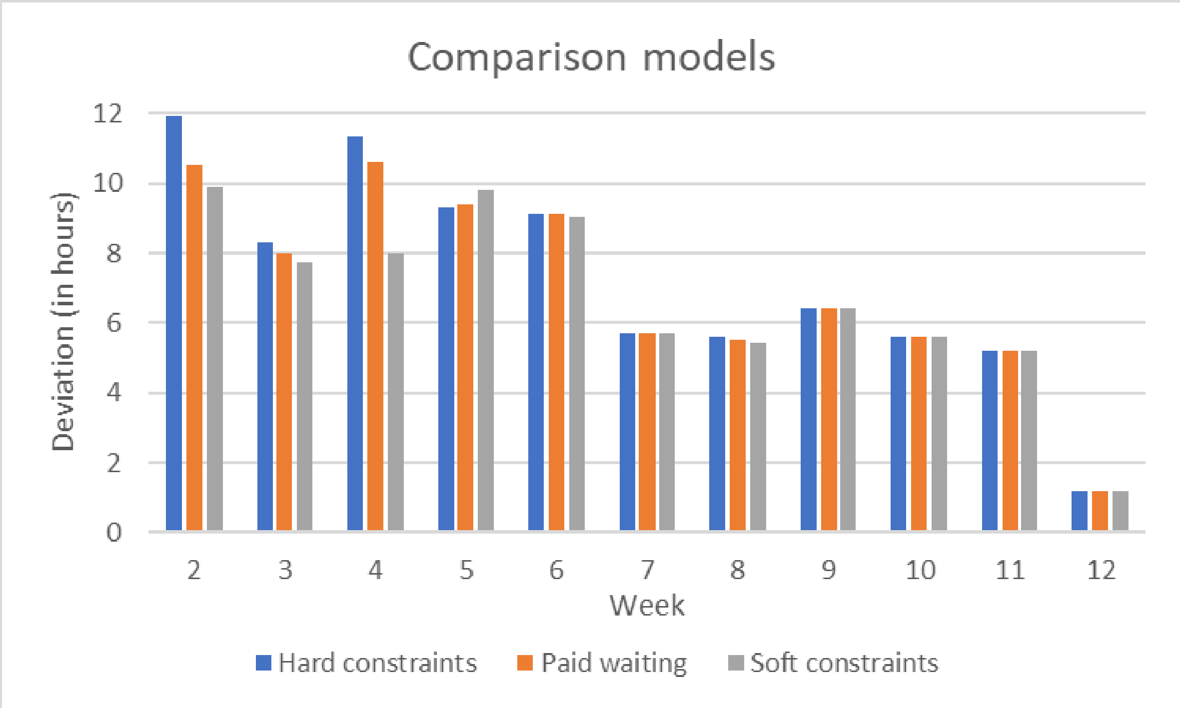

Figure 23: Comparison performance models

As the figure shows, there is no model that specifically performs the best in all situations within the maximum running time of 3600 seconds. However, some observations can be highlighted. The first is that both the model with the paid waiting time and the model with soft constraints can find a feasible solution for scenario 12, where the model with hard constraints could not without adjustments.

Next, both the model with the paid waiting time and the soft constraints did not always get to the optimal solution within the running time of 3600 seconds. Specifically we see that in instance 5 the model with hard constraints finds a better solution than both the model with paid waiting time and the model with soft constraints. This is since the models with paid waiting time and soft constraints are not solved to optimality and ends up with gaps of 1.2% and 5.9% respectively. This is due to the larger solution space compared to the model with hard constraints, these models need more time to explore all possible solutions to come to the optimal solution.

On average, the model with soft constraints performs the best, with 6.74 hours of deviation. However, it is important to note that this lower deviation between scheduled and contractual hours comes with a penalty cost. This penalty cost is 0.2 hours violation per week on average (Table 12).

5.8. IBM CPLEX vs Python MIP

For all experiments we used IBM CPLEX as solver. IBM CPLEX is a license-based commercial solver and thus not free. In this section we compare the license-based commercial solver with a free solver. The free solver we use is Python MIP. The reason behind the comparison is for the company to decide whether the license-based solver is worth the investment. For example, when the free solver performs equally, it is not worth to invest in a license for a license-based solver.

The model we use for this comparison is the model containing soft constraints regarding the start times (Section 5.5). This model is able to find feasible solutions due to the possibility to violate constraints and performs on average better than the model where we allow paid waiting time.

Since CPLEX is a license-based commercial solver and Python MIP is a free solver, we expect Python MIP to take more running time to solve the problem to approximately the same solution value. This is the reason that we compare the time until the first solution is found for both solvers.

Scenario	CPLEX			Python MIP		
	Running time	Solution value	GAP	Running time	Solution value	GAP
2	8	249.3	97.0%	195	54.2	86.3%
3	6	247.2	97.4%	308	36.3	81.5%
4	9	243.0	96.5%	153	39.5	78.8%
5	4	237.8	97.3%	162	31.0	74.4%
6	3	219.5	96.9%	180	43.0	81.2%
7	2	305.1	99.8%	201	55.8	90.8%
8	6	284.1	98.2%	200	40.4	88.0%
9	4	261.8	98.6%	162	25.8	77.5%
10	7	220.2	98.1%	202	60.4	92.6%
11	5	229.6	98.2%	147	24.4	79.9%
12	4	189.5	99.3%	61	16.3	92.0%
Average	5	244.3	96.5%	179	38.8	83.9%

Table 13: First solution found CPLEX vs Python MIP

Table 13 shows the differences in running times and solution values between CPLEX and Python MIP. There are some remarkable differences between the results. The running times of CPLEX are significantly lower than the running times of Python MIP. The average running times for CPLEX is 5 seconds versus 179 seconds for Python MIP. On the other hand, the first found solution found with Python MIP is significantly better than the first solution found with CPLEX. CPLEX finds an average solution value of 244.3, where MIP finds an average of 38.8. The reason Python MIP finds a better first solution than CPLEX can be caused by the strategy used in the solver. CPLEX is focused on finding a feasible solution fast, however, quality seems to be less important for this first solution. On the other hand Python MIP is focused on finding good quality solutions as a first solution, which makes it logical that it takes more time to find a first solution compared to CPLEX.

So in terms of finding solutions quickly, CPLEX outperforms Python MIP as expected. However, the first solutions found using Python MIP are already lower than the current performance of the transport planners. To add up on that, it has to be kept in mind that this only concerns a tactical schedule. So since the schedule is not made daily, longer running times are acceptable.

Since the first solution found is not the best solution, we are not specifically interested in when the first solution is found. We are interested in which solver performs the best after a certain running time. The running times we use are 5, 10, 15, 20, 30, 45 and 60 minutes. Figure 24 shows the average optimality gaps found after a certain running time per solver.

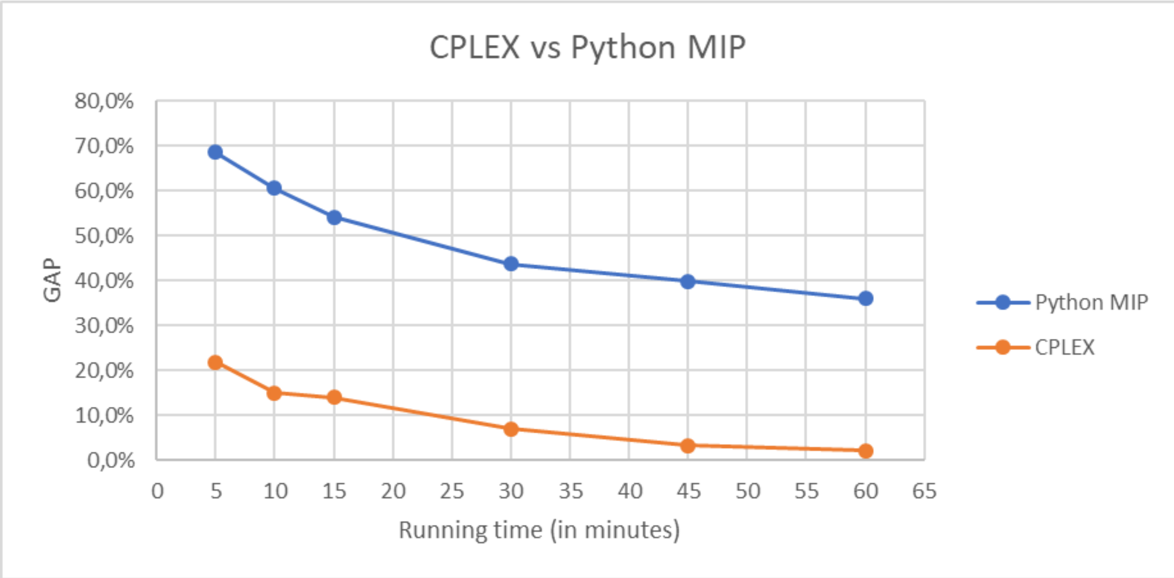

Figure 24: Optimality gaps found after x running times (comparison CPLEX vs Python MIP)

When looking at the figure, we see that CPLEX outperforms Python MIP significantly. CPLEX finds an average gap of 23% in 5 minutes while Python MIP takes 60 minutes to find an average gap of 36%.

5.9. Analysis 2020

To give a good comparison between the current situation where everything is done manually and the situation where the solver is used, we use data of the year 2020. We compare the quality of current schedules with the quality of the proposed schedules by the solver. For this comparison, we use the CPLEX solver. This is done since the running times are less important when comparing the solver with the old schedules.

In order to compare the solver with all weeks of 2020, we first need to analyze all weeks of 2020 in terms of deviation between scheduled and contractual hours. This analysis is done in the same way we did in Section 2.4. The results of the analysis are presented in Appendix C. The average total deviation per week in 2020 is 147.9 hours.

We execute all 3 models from Section 5.4, 5.5 and 5.6. We use each week of 2020 as a scenario and also execute the scenarios with a maximum running time of 3600 seconds. Below we show the averages of the results of the analysis. Appendix C contains all results of all experiments.

Hard constraints

	Current situation	Hard constraints			
	Total deviation (in hours)	**Total deviation (in hours)**	**Reduction (in %)**	**GAP (in %)**	**Running time (in seconds)**
Average	147.9	5.4	96.3%	0.0%	302

Table 14: Results analysis 2020 with hard constraints

Table 14 shows the results of the analysis of 2020 where we consider all constraints to be hard constraints. The model with hard constraints realizes an average total deviation of 5.4 hours per week, which is a reduction of 96.3% in deviation between scheduled and contractual hours. However, it is important to note that there does not exist a feasible solution for 14 of the 53 scenarios. All (feasible) instances are solved to optimality within the maximum running time of 3600. The average running time is 300 seconds.

Paid waiting time

	Current situation	Paid waiting			
	Total deviation (in hours)	**Total deviation (in hours)**	**Reduction (in %)**	**GAP (in %)**	**Running time (in seconds)**
Average	147.9	7.2	95.2%	0.5%	642

Table 15: Results analysis 2020 with allowing paid waiting time

Table 15 shows the results of the option where we allow paid waiting time. The model where we allow paid waiting time gives us an average total deviation of 7.2 hours reduction between scheduled and contractual hours. This is a reduction of 95.2% compared to the current situation. Not all instances solve to optimality. 5 of the 53 instances do not solve to optimality within the maximum running time of 3600 seconds, resulting in an overall average gap of 0.5%. The average running time is 642 seconds.

Soft constraints

	Current situation	Soft constraints				
Week	**Total deviation (in hours)**	**Total deviation (in hours)**	**Penalty**	**Reduction (in %)**	**GAP (in %)**	**Running time (in seconds)**
Average	147.9	6.5	0.5	95.7%	0.1%	566

Table 16: Results analysis 2020 with soft constraints

Table 16 shows the results of the model where the model has the option to violate the restrictions regarding the start times. The model with soft constraints realizes an average total deviation of 6.5 hours between scheduled and contractual hours, which is a reduction of 95.7% compared to the current situation. However, this reduction comes with an average penalty of 0.5 hours per week. This means that on average the restrictions on the start times are violated 0.5 hours. 3 of the 53 scenarios do not solve to optimality within the maximum running time of 3600 seconds, resulting in an average gap of 0.1%. The average running time for the model with soft constraints is 566 seconds.

Over the whole year 2020, the model with hard constraints seems to have the biggest reduction of 96.3% compared to the current situation. However, this model has the disadvantage that there does not always exist a feasible solution. So, the model that performs the best in terms of both finding feasible solutions and realizing the biggest reduction, is the model with soft constraints. This model realizes a reduction of 95.7%.

From both the comparison in Section 5.6 and the analysis of 2020, we conclude that the model with soft constraints is the model that performs the best in terms of a minimal deviation between scheduled and contractual hours.

5.10. Sensitivity analysis

As mentioned in Section 5.2, we carry out a Monte Carlo simulation. We let the parameter $TotalTime_j$ be a stochastic variable. We use real data to simulate the stochasticity of the parameter to get a new total shift time. This new total time of a shift represents the execution time it takes. With the new total shift time, we calculate the overtime or undertime when the shifts are executed.

We decide to evaluate the robustness of the rosters that are already created. We generate 1000 instances per already created weekly schedule for weeks 2 to 12 of 2021. The outcome of the analysis is the deviation between contractual and (simulated) executed hours. The boxplot in Figure 25 shows the results of the Monte Carlo simulation of each week for the situation with hard constraints.

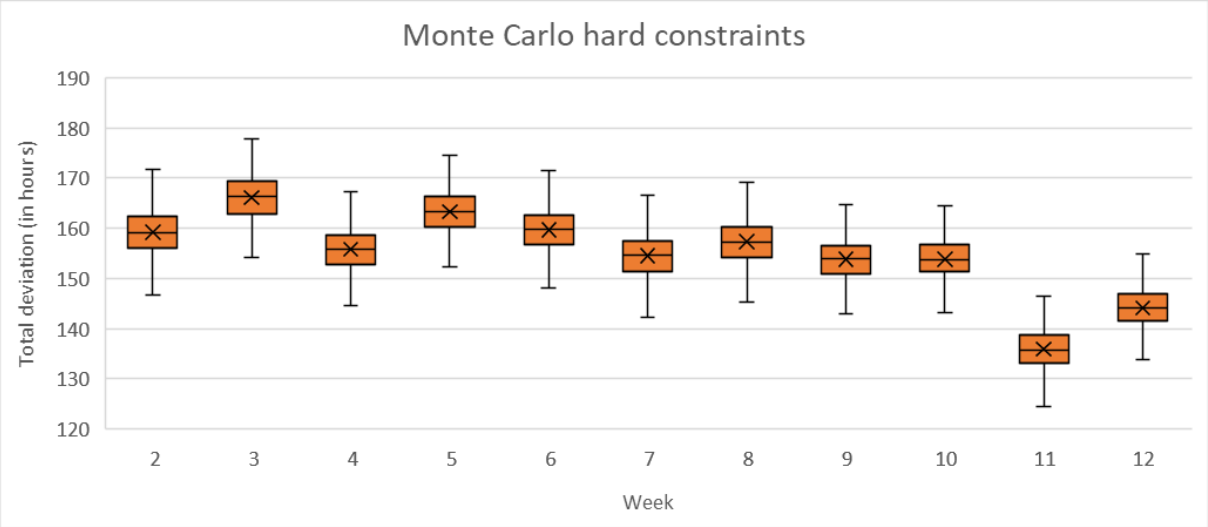

Figure 25: Monte Carlo simulation hard constraints

Figure 26 shows the results for the Monte Carlo simulation in the case where we allow paid waiting time.

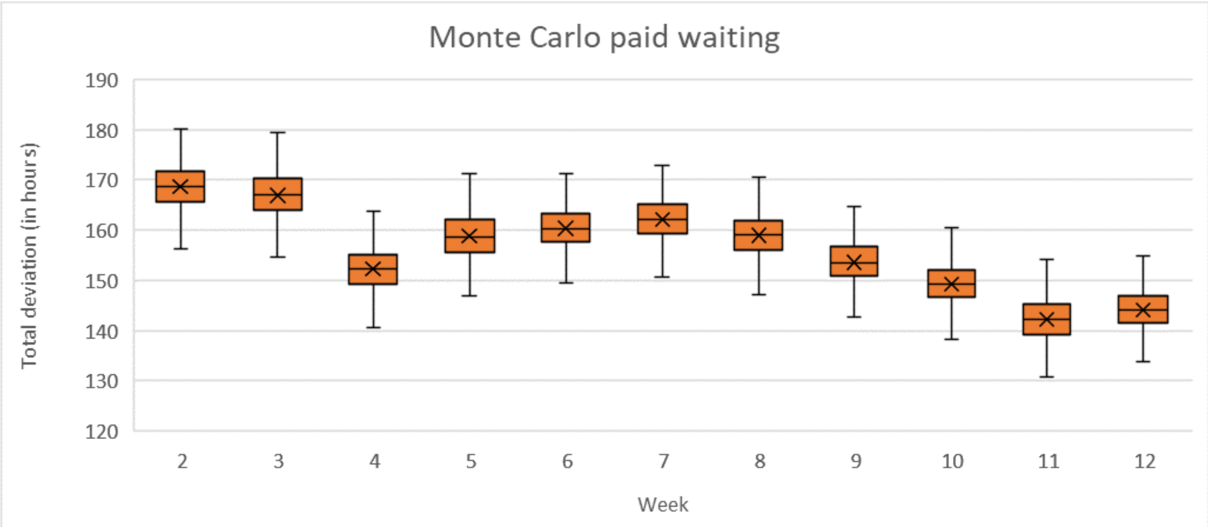

Figure 26: Monte Carlo simulation paid waiting time

Figure 27 shows the results for the Monte Carlo simulation in the case where we use soft constraints for the start times.

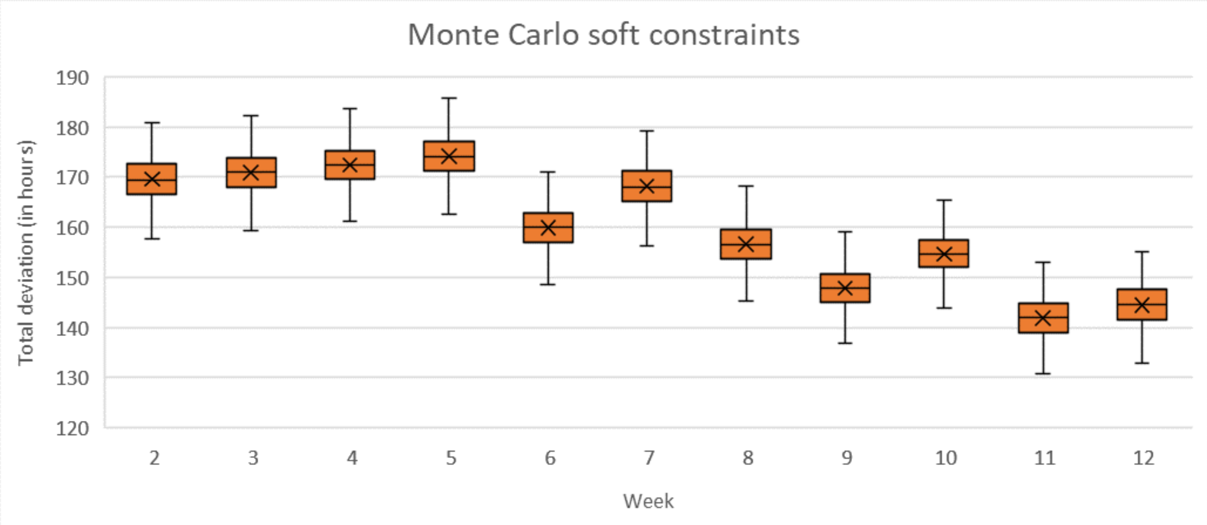

Figure 27: Monte Carlo simulation soft constraints

In all figures we see that running the Monte Carlo simulation results in a higher deviation between contractual and worked/scheduled hours (between 130 and 180 hours). This is logical since the real data shows that execution takes on average 7% more time than scheduled. All working hours are included in the executed hours. So, also unplanned waiting time due to traffic jams are included. These waiting time are not scheduled, but do count towards the executed hours.

The average deviation over all weeks after execution is for the model with hard constraints 154 hours, for the model with paid waiting time 156 hours and with soft constraints 160 hours. This little difference can be caused by the fact that the model with soft constraints had the lowest deviation between scheduled and contractual hours. When it takes on average 7% more hours to execute a schedule than planned, the model with soft constraints will result in more deviation, since it had less undertime scheduled. The same comparison is applicable to the model where we allow paid waiting time. In this model we have less undertime than the model with hard constraints, resulting is more overtime after execution.

In order to create a schedule that does not contain a lot of overtime after execution, the scheduled hours should be lower than the contractual hours or the estimation of the duration of shifts should be more precise.

5.11. Incorporation of 46-hour contracts

As mentioned in Section 5.2, the feeling exists that the deviation is caused more by specific drivers. So, in this section we dive into the content of the schedules. Instead of using the objective value function, we look at the deviation per driver. With this analysis we see if the objective value is influenced more by specific drivers. This influence from specific drivers can have a relation with the restrictions or contractual hours of these drivers.

The most remarkable thing in the undertime hours is that 2 specific drivers negatively influence the objective value the most. The average undertime hours over week 2 to 12 is on average 2.16 for the 2 drivers separately. This means that the 2 drivers together result in an average of 4.32 hours of deviation each week. Table 17 shows that the 2 drivers are accountable for 59.2% of the solution value on average.

Model	Average percentage of solution value
Hard constraints	57.3%
Paid waiting time	60.2%
Soft constraints	60.3%
Average	59.2%

Table 17: Percentage of solution value for 46 hour contracts

These drivers have a contract saying they have to work 46 hours (excluding breaks). These 46 hours have to be spread over 4 working days. This means that those drivers have to work 11.5 hours on average per day, excluding breaks. Since those shifts do not exist/are planned by the Supply Chain Planner, the schedule will result in undertime for the 2 mentioned drivers.

5.12. Conclusions

This chapter answered the following research question: "How does the method perform (compared to the current situation)?"

First, we included extra constraints to reflect real rosters as good as possible. These extra constraints did influence the solution value, but the model still performs better than the current performance. We tested 3 models: consider all constraints as hard constraints, allowing paid waiting time and consider the restrictions regarding the start times as soft constraints. The model with hard constraints has the disadvantage that it cannot always find a solution due to infeasibility. The other two models were always able to find a solution. The model containing soft constraints performs the best on average compared to the other 2 models.

We compared a commercial license-based solver (IBM CPLEX) with a free solver (Python MIP). Here we saw that the license-based solver outperforms the free solver. Within 5 minutes of running time, the license-based solver finds better solutions than the free solver does in 1 hour. Our analysis of 2020 confirms that the model with soft constraints realizes the biggest reduction in deviation between scheduled and contractual hours.

The sensitivity analysis shows us that the execution of each schedule results in a total overtime between 130 and 180 hours. This is due to the fact that the execution adds an average of 7% to the scheduled hours. Finally, we analysed the content of the schedules and conclude that 2 specific drivers influence the solution in a negative way the most. These drivers have a contract indicating 46 working hours. The influence of these 2 drivers is on average 59.2% on the solution value.

6. Conclusions and recommendations

Section 6.1 presents the drawn conclusions based on the research. We present the recommendations in Section 6.2. Section 6.3 discusses possibilities for further research.

6.1. Conclusions

In this section, we draw conclusions based on the research carried out at the company.

The main research question is formulated as follows:

> ***"What method should the company use to create a weekly schedule for its drivers that can be used along with the advanced planning system with the aim of lowest cost possible?"***

Firstly, we analysed the current situation. The company does not use any scheduling tools to schedule their drivers. The current process of scheduling drivers is a manual process which makes it a time-consuming and failure sensitive process. It takes 48 hours to build a weekly schedule from scratch. The result is a schedule that contains a lot of deviation between scheduled and contractual hours, on average 2.43 hours per driver per week.

Next, we performed a literature review regarding scheduling personnel. We introduced general problems. Our problem had overlap with specific problems, these were: the nurse scheduling problem, the airline crew scheduling problem, the bus driver rostering problem and the set partitioning problem. Regarding the optimization methods, heuristics are used for NP-hard problems to find good solutions fast. However, also mathematical models are used to model and solve scheduling problems. To our best knowledge, the literature lacks a set partitioning problem where not all shifts have to be filled.

A mathematical model is formulated to improve the scheduling process. The advantage of the model is that it creates a schedule with less deviation between scheduled and contractual hours in less time than the operational transport planners. Pre-assigning several decision variables by the use of pre-processing techniques helps to reduce the problem size and thus the running time. Since the method should be used along with the advanced planning system, the output of the system functions as input for our model.

We tested our model with 11 instances under different circumstances. Firstly, we included extra constraints to make a more likeable schedule for the drivers. Then we compared this model with the model containing paid waiting time and the model with soft constraints. The model with hard constraints performed better than the current situation, but cannot always find a feasible solution. There does not exist a solution that satisfies all constraints if only hard constraints are used. To solve that, we propose models with paid waiting time and soft constraints that find feasible solutions for each scenario. We performed all experiments with a running time of 3600 seconds. On average the model with soft constraints performs best. The average reduction in deviation between scheduled and contractual hours compared to the current situation is 95.8% (Table 18). However, this reduction comes with an average penalty of 0.2 hours.

Model	Average reduction compared to current situation
Hard constraints	95.5%
Paid waiting time	95.6%
Soft constraints	95.8%

Table 18: Performance models

Our simulation of 2020 confirms our findings that the model with soft constraints indeed performs the best of all 3. We analyzed the robustness of the schedules using a Monte Carlo simulation. The simulation shows that the realized hours are 7% higher than the scheduled hours, resulting in a high deviation between worked and contractual hours. A few drivers have a large influence on the solution value. We showed that the drivers with a contract of 46 hours (excluding breaks) care for 59.2% of the total deviation between scheduled and contractual hours. We also showed that a commercial license-based solver performs significantly better than a free solver.

In Section 1.4.3 the norm is stated that at least the same quality schedules can be created in half the effort it takes in the current situation. When using the method, we create better schedules in less than 1 hour (compared to 48 labor hours in the current situation). So we create faster a better schedule, which means that the norm is achieved.

The schedules are better in terms of deviation, but also in terms of failure sensitivity. Since the problem is approached mathematically and solved by a computer, no failures are made if the input is right.

6.2. Recommendations

In order to improve the process of scheduling drivers, the company can improve by the following recommendations:

- **Start using the model with soft constraints as a tool to help the planner in making weekly driver schedules**

Our first recommendation is to start using the model as a tool to help the planner in making weekly driver schedules. The model that performs the best on average was the model with soft constraints regarding the start times, this model is thus also recommended to use.

We emphasize that both the model and the planner should be used in their strengths. The strength of the model is the computational power to make rosters that minimize the cost. The strength of the planner is the experience and human intuition to handle with uncommon situations.

The running time can be set to 1 hour. Since the schedules that are created are weekly schedules, longer running times are affordable. When the model finds an optimal solution within this hour, the calculations complete earlier.

Implementing the model results in a reduction of the time it takes to create a schedule and a reduction in deviation between scheduled and contractual hours. The implementation realizes a reduction of 48 labor hours per schedule created. The reduction in deviation between scheduled and contractual hours is 95.8% on a tactical level.

It is important to note that changes in the model require knowledge of mathematical modelling. So it is recommended to educate employees in mathematical modelling. When these employees are educated, they can adjust the model when needed or apply this knowledge to other problems within the company.

- **Drivers with 46 contractual hours**

We recommend making adjustments regarding the drivers with 46-hour contracts. Drivers with a contract of 46 hours have a large influence on the deviation between scheduled and contractual hours. As mentioned in Section 5.11, they care for an average of 59.2% of the deviation between scheduled and contractual hours. The 46 hours have to be spread over 4 working days, which result in long shifts. These long shifts are not provided by the Supply Chain Planner. In order to reduce the large influence of the contracts with 46 hours, there are 3 options:

The company can offer to reduce the contract to 40 hours. When the driver accepts to have a working week of 40 hours, it is easier to reach the contractual hours of the drivers. This results in a reduction in undertime of these drivers and thus a reduction in total deviation between scheduled and contractual hours.

The company can offer the drivers to have 5 workings days. When the drivers work 5 days per week instead of 4, the driver logically works 5 shifts. It is easier to reach 46 working hours in 5 shifts than in 4 shifts.

The Supply Chain Planner can provide longer shifts. When the Supply Chain Planner provides longer shifts for the drivers with a 46-hour contract, it reduces the undertime. However, when the drivers are not present (due to holiday or sickness), another driver (or temporary employee or external transporters) has to complete these longer shifts.

- **Improving schedule robustness**

Section 5.10 shows that the actual time spend is always higher than scheduled. The average of 7% extra hours that are added during execution result in a total overtime between 130 and 180 hours. To make the schedule more robust and come closer to the contractual hours, the company has 2 options:

When the company schedules fewer hours than the contractual hours, it avoids overtime. In this way, there exists slack for shifts that take longer than scheduled. The disadvantage of scheduling standard undertime is that the undertime has to be paid when it still exists after execution.

Another option to avoid overtime is by improving the estimation of the shift durations. When the estimation of the shift duration is done more exactly and there is less deviation in duration between scheduled and executed shifts, the overtime reduces.

6.3. Future research

First, the input for the model should be right. It is failure-proof to maintain the master data in a system within the company. By using this system, the master data is complete and maintainable. The export of this system is input for our model. In this way, the company is documenting the right data, all together, in the right way.

Currently, the company uses paper notes to communicate the schedules to the driver. This is old-fashioned and out of date. It can be improved by digitalizing the schedules. In this way, the drivers can find their schedules online and possible changes can also be shared online.

The developed model is meant to use for the weekly schedule. This means that daily operational changes still have to be done manually. Further research can be done in automating the daily scheduling process as well. This reduces the failure sensitivity and time needed to complete the daily schedules after operational changes.

Appendix A: Bibliography

Aickelin, U., & Dowsland, K. A. (2004). An indirect Genetic Algorithm for a nurse-scheduling problem. *Computers & Operations Research*, *31*(5), 761–778. https://doi.org/10.1016/s0305-0548(03)00034-0

Akbarzadeh, B., & Maenhout, B. (2021). An exact branch-and-price approach for the medical student scheduling problem. *Computers & Operations Research*, *129*, 105209. https://doi.org/10.1016/j.cor.2021.105209

Arkin, E., Tekinerdogan, B., & İmre, K. M. (2016). Systematic approach for deriving feasible mappings of parallel algorithms to parallel computing platforms. *Concurrency and Computation: Practice and Experience*, *29*(1). https://doi.org/10.1002/cpe.3821

Bach, L., Dollevoet, T., & Huisman, D. (2016). Integrating Timetabling and Crew Scheduling at a Freight Railway Operator. *Transportation Science*, *50*(3), 878–891. https://doi.org/10.1287/trsc.2015.0648

Barbosa, V., Alvelos, F., & Respício, A. (2013). A Hybrid Metaheuristic for the Bus Driver Rostering Problem. *Proceedings of the 2nd International Conference on Operations Research and Enterprise Systems*. https://doi.org/10.5220/0004283700320042

Barbosa, V., Respício, A., & Alvelos, F. (2015). A Column Generation Based Heuristic for a Bus Driver Rostering Problem. *Progress in Artificial Intelligence*, 143–156. https://doi.org/10.1007/978-3-319-23485-4_16

Boschetti, M. A., Mingozzi, A., & Ricciardelli, S. (2008). A dual ascent procedure for the set partitioning problem. *Discrete Optimization*, *5*(4), 735–747. https://doi.org/10.1016/j.disopt.2008.06.001

Cappanera, P., & Gallo, G. (2004). A Multicommodity Flow Approach to the Crew Rostering Problem. *Operations Research*, *52*(4), 583–596. https://doi.org/10.1287/opre.1040.0110

Clausen, J. (1999). *Branch and Bound Algorithms - Principles and Examples*. University of Copenhagen.

Dawid, H., König, J., & Strauss, C. (2001). An enhanced rostering model for airline crews. *Computers & Operations Research*, *28*(7), 671–688. https://doi.org/10.1016/s0305-0548(00)00002-2

Daş, G. S., Altınkaynak, B., Göçken, T., & Türker, A. K. (2021). A set partitioning based goal programming model for the team formation problem. *International Transactions in Operational Research*, *29*(1), 301–322. https://doi.org/10.1111/itor.13022

Dowsland, K. A. (1998). Nurse scheduling with tabu search and strategic oscillation. *European Journal of Operational Research*, *106*(2-3), 393–407. https://doi.org/10.1016/s0377-2217(97)00281-6

Er-Rbib, S., Desaulniers, G., El Hallaoui, I., & Bani, A. (2020). Integrated and sequential solution methods for the cyclic bus driver rostering problem. *Journal of the Operational Research Society*, *72*(4), 764–779. https://doi.org/10.1080/01605682.2019.1700187

Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, *153*(1), 3–27. https://doi.org/10.1016/s0377-2217(03)00095-x

Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B., & Sier, D. (2004). An Annotated Bibliography of Personnel Scheduling and Rostering. *Annals of Operations Research*, *127*(1-4), 21–144. https://doi.org/10.1023/b:anor.0000019087.46656.e2

Ezzinbi, O., Sarhani, M., El Afia, A., & Benadada, Y. (2014). Particle swarm optimization algorithm for solving airline crew scheduling problem. *2014 International Conference on Logistics Operations Management*. https://doi.org/10.1109/gol.2014.6887447

Farasat, A., & Nikolaev, A. G. (2016). Signed social structure optimization for shift assignment in the nurse scheduling problem. *Socio-Economic Planning Sciences*, *56*, 3–13. https://doi.org/10.1016/j.seps.2016.06.003

Freling, R., Lentink, R. M., & Wagelmans, A. P. M. (2004). A Decision Support System for Crew Planning in Passenger Transportation Using a Flexible Branch-and-Price Algorithm. *Annals of Operations Research*, *127*(1-4), 203–222. https://doi.org/10.1023/b:anor.0000019090.39650.32

Friedrich, C., & Elbert, R. (2022). Adaptive large neighborhood search for vehicle routing problems with transshipment facilities arising in city logistics. *Computers & Operations Research*, *137*, 105491. https://doi.org/10.1016/j.cor.2021.105491

Gamache, M., & Soumis, F. (1998). A Method for Optimally Solving the Rostering Problem. *Operations Research in the Airline Industry*, 124–157. https://doi.org/10.1007/978-1-4615-5501-8_5

Glover, F., & Laguna, M. (2002). *Tabu search*. Kluwer Academic Publishers.

Hadwan, M., & Ayob, M. (2010). A constructive shift patterns approach with simulated annealing for nurse rostering problem. *2010 International Symposium on Information Technology*. https://doi.org/10.1109/itsim.2010.5561304

Hasebe, M., Yamazaki, T., Ryumae, M., Wu, W., Nonobe, K., & Ikegami, A. (2017). A comparison of integer programming formulations and variable-fixing method for the nurse scheduling problem. *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. https://doi.org/10.1109/ieem.2017.8289853

Heerkens, H., & Van Winden, A. (2017). *Solving Managerial Problems Systematically*. Noordhoff Uitgevers.

Horváth, M., & Kis, T. (2017). Computing strong lower and upper bounds for the integrated multiple-depot vehicle and crew scheduling problem with branch-and-price. *Central*

European Journal of Operations Research, *27*(1), 39–67. https://doi.org/10.1007/s10100-017-0489-4

Ikegami, A., & Niwa, A. (2003). A subproblem-centric model and approach to the nurse scheduling problem. *Mathematical Programming*, *97*(3), 517–541. https://doi.org/10.1007/s10107-003-0426-2

Ishibuchi, H., Hitotsuyanagi, Y., Tsukamoto, N., & Nojima, Y. (2008). Use of Heuristic Local Search for Single-Objective Optimization in Multiobjective Memetic Algorithms. *Parallel Problem Solving from Nature – PPSN X*, 743–752. https://doi.org/10.1007/978-3-540-87700-4_74

J. Lim, G., Mobasher, A., & J. Côté, M. (2012). Multi-objective Nurse Scheduling Models with Patient Workload and Nurse Preferences. *Management*, *2*(5), 149–160. https://doi.org/10.5923/j.mm.20120205.03

Joseph, A. (2002). A concurrent processing framework for the set partitioning problem. *Computers & Operations Research*, *29*(10), 1375–1391. https://doi.org/10.1016/s0305-0548(01)00037-5

Kheiri, A., Gretsista, A., Keedwell, E., Lulli, G., Epitropakis, M. G., & Burke, E. K. (2021). A hyper-heuristic approach based upon a hidden Markov model for the multi-stage nurse rostering problem. *Computers & Operations Research*, *130*, 105221. https://doi.org/10.1016/j.cor.2021.105221

Knust, F., & Xie, L. (2017). Simulated annealing approach to nurse rostering benchmark and real-world instances. *Annals of Operations Research*, *272*(1-2), 187–216. https://doi.org/10.1007/s10479-017-2546-8

Legrain, A., Bouarab, H., & Lahrichi, N. (2014). The Nurse Scheduling Problem in Real-Life. *Journal of Medical Systems*, *39*(1). https://doi.org/10.1007/s10916-014-0160-8

Leung, P. P. L., Wu, C. H., Kwong, C. K., Ip, W. H., & Ching, W. K. (2021). Digitalisation for optimising nursing staff demand modelling and scheduling in nursing homes. *Technological Forecasting and Social Change*, *164*, 120512. https://doi.org/10.1016/j.techfore.2020.120512

Li, X., Chen, C., Luo, Y., & Chen, M. (2017). Optimization Scheme Based on Parallel Computing Technology. *Communications in Computer and Information Science*, 504–513. https://doi.org/10.1007/978-981-10-6442-5_48

Limlawan, V., Kasemsontitum, B., & Jeenanunta, C. (2011). Airline crew rostering problem using particle swarm optimization. *2011 IEEE International Conference on Quality and Reliability*. https://doi.org/10.1109/icqr.2011.6031590

Lin, C.-H. M., & Salkin, H. M. (1983). An efficient algorithm for the complete set partitioning problem. *Discrete Applied Mathematics*, *6*(2), 149–156. https://doi.org/10.1016/0166-218x(83)90069-0

Lučic, P., & Teodorovic, D. (1999). Simulated annealing for the multi-objective aircrew rostering problem. *Transportation Research Part A: Policy and Practice*, *33*(1), 19–45. https://doi.org/10.1016/s0965-8564(98)00021-4

Maenhout, B., & Vanhoucke, M. (2010). A hybrid scatter search heuristic for personalized crew rostering in the airline industry. *European Journal of Operational Research*, *206*(1), 155–167. https://doi.org/10.1016/j.ejor.2010.01.040

Morris, J. G., & Showalter, M. J. (1983). Simple Approaches to Shift, Days-Off and Tour Scheduling Problems. *Management Science*, *29*(8), 942–950. https://doi.org/10.1287/mnsc.29.8.942

Müller, T. (1998). Solving Set Partitioning Problemswith Constraint Programming. *Proceedings of the Sixth International Conference on the Practical Application of Prolog and the Fourth International Confernce on the Practical Application of Constraint Technology*, 313–332.

Ni, H., & Abeledo, H. (2007). A branch-and-price approach for large-scale employee tour scheduling problems. *Annals of Operations Research*, *155*(1), 167–176. https://doi.org/10.1007/s10479-007-0212-2

Nishi, T., Muroi, Y., & Inuiguchi, M. (2011). Column generation with dual inequalities for railway crew scheduling problems. *Public Transport*, *3*(1), 25–42. https://doi.org/10.1007/s12469-011-0037-x

Osogami, T., & Imai, H. (2000). Classification of Various Neighborhood Operations for the Nurse Scheduling Problem. *Algorithms and Computation*, 72–83. https://doi.org/10.1007/3-540-40996-3_7

Oughalime, A., Ismail, W. R., & Yeun, L. C. (2008). A Tabu Search approach to the nurse scheduling problem. *2008 International Symposium on Information Technology*. https://doi.org/10.1109/itsim.2008.4631604

Pan, S., Akplogan, M., Touati, N., Létocart, L., Wolfler Calvo, R., & Rousseau, L.-M. (2018). A hybrid heuristic for the multi-activity tour scheduling problem. *Electronic Notes in Discrete Mathematics*, *69*, 333–340. https://doi.org/10.1016/j.endm.2018.07.043

Peng, X., Chen, S., Jin, H., Ma, Z., & Jia, W. (2016). A Two-Stage Model of Crew Rostering Considering the Balance of Working Intensity for Urban Rail Transit. *CICTP 2016*. https://doi.org/10.1061/9780784479896.089

Perumal, S. S. G., Larsen, J., Lusby, R. M., Riis, M., & Sørensen, K. S. (2019). A matheuristic for the driver scheduling problem with staff cars. *European Journal of Operational Research*, *275*(1), 280–294. https://doi.org/10.1016/j.ejor.2018.11.011

Respício, A., Moz, M., & Vaz Pato, M. (2013). Enhanced genetic algorithms for a bi-objective bus driver rostering problem: a computational study. *International Transactions in Operational Research*, *20*(4), 443–470. https://doi.org/10.1111/itor.12013

Rong, A. (2010). Monthly tour scheduling models with mixed skills considering weekend off requirements. *Computers & Industrial Engineering*, *59*(2), 334–343. https://doi.org/10.1016/j.cie.2010.05.005

Rushmeier, R., Hoffman, K., Padberg, M. (1995). Recent Advances in Exact Optimization of Airline Scheduling Problems. *Airline scheduling*,

Savelsbergh, M. W. (1994). Preprocessing and Probing Techniques for Mixed Integer Programming Problems. *ORSA Journal on Computing*, *6*(4), 445–454. https://doi.org/10.1287/ijoc.6.4.445

van Krieken, M. G. C. (2006). Solving set partitioning problems using lagrangian relaxation. CentER, Center for Economic Research.

Xie, L., Merschformann, M., Kliewer, N., & Suhl, L. (2017). Metaheuristics approach for solving personalized crew rostering problem in public bus transit. *Journal of Heuristics*, *23*(5), 321–347. https://doi.org/10.1007/s10732-017-9348-7

Xue, N., Landa-Silva, D., Triguero, I., & Figueredo, G. P. (2018). A Genetic Algorithm With Composite Chromosome for Shift Assignment of Part-time Employees. *2018 IEEE Congress on Evolutionary Computation (CEC)*. https://doi.org/10.1109/cec.2018.8477818

Zhang, Y., Rao, Y., & Zhou, M. (2007). GASA hybrid algorithm applied in airline crew rostering system. *Tsinghua Science and Technology*, *12*(S1), 255–259. https://doi.org/10.1016/s1007-0214(07)70120-7

Appendix B: Experimental setting

Number of drivers present per day

Number of drivers present per day

Week	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
2	40	45	42	43	42	24	4
3	43	45	41	41	40	22	3
4	38	40	39	41	38	17	3
5	45	45	43	38	37	21	1
6	40	39	38	41	41	22	3
7	40	41	43	37	34	19	2
8	41	40	41	41	36	20	2
9	44	40	42	34	31	21	1
10	36	41	41	36	39	19	3
11	39	36	37	31	32	20	1
12	33	33	32	36	38	20	3

Number of shifts per day

Number of shifts per day

Week	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
2	84	87	87	97	101	94	14
3	85	88	88	98	102	95	14
4	85	88	88	98	102	95	14
5	85	88	88	98	102	95	14
6	85	88	88	98	102	95	14
7	85	88	88	98	102	95	14
8	85	88	88	98	102	95	14
9	85	88	88	98	102	95	14
10	85	88	88	98	102	95	14
11	85	88	88	98	102	95	14
12	85	91	91	102	107	98	14

Appendix C: Comparison 2020

Hard constraints

	Current situation	Hard constraints			
Week	**Total deviation (in hours)**	**Total deviation (in hours)**	**Reduction (in %)**	**GAP (in %)**	**Running time**
1	112.8	1.3	99%	0%	49
2	101.0	1.2	99%	0%	39
3	161.9	Infeasible			
4	157.4	Infeasible			
5	154.3	2.4	98%	0%	534
6	157.1	3.9	98%	0%	1418
7	132.3	Infeasible			
8	146.5	Infeasible			
9	146.4	1.9	99%	0%	777
10	227.4	Infeasible			
11	136.1	10.4	92%	0%	243
12	150.6	Infeasible			
13	119.0	Infeasible			
14	140.4	Infeasible			
15	183.7	24.7	87%	1%	1801
16	143.2	8.3	94%	0%	144
17	155.9	Infeasible			
18	167.2	5.3	97%	0%	216
19	151.4	4.3	97%	0%	299
20	144.4	Infeasible			
21	136.8	4.6	97%	0%	919
22	129.0	5.3	96%	0%	387
23	137.3	7.9	94%	0%	367
24	172.1	10.3	94%	0%	191
25	156.1	Infeasible			
26	147.3	Infeasible			
27	147.7	Infeasible			
28	109.0	6.9	94%	0%	150
29	108.5	7.3	93%	0%	36
30	124.5	1.4	99%	0%	86
31	115.2	6.0	95%	0%	163
32	132.5	8.2	94%	0%	310
33	122.3	7.1	94%	0%	55
34	132.8	8.4	94%	0%	449
35	130.6	2.1	98%	0%	96
36	173.2	Infeasible			
37	129.7	3.4	97%	0%	123
38	165.8	7.5	95%	0%	188

39	154.5	10.3	93%	0%	110
40	175.1	3.6	98%	0%	149
41	150.7	5.7	96%	0%	139
42	171.4	3.4	98%	0%	124
43	153.4	4.3	97%	0%	123
44	208.1	0.4	100%	0%	412
45	169.7	3.9	98%	0%	156
46	172.9	3.8	98%	0%	292
47	144.9	3.7	97%	0%	153
48	150.9	6.6	96%	0%	139
49	168.9	3.5	98%	0%	289
50	165.9	4.7	97%	0%	236
51	152.9	3.6	98%	0%	216
52	133.8	2.9	98%	0%	147
53	106.5	1.1	99%	0%	36
Average	**147.9**	**5.4**	**96.3%**	**0.0%**	**302**

Paid waiting time

	Current situation	**Paid waiting**			
Week	**Total deviation (in hours)**	**Total deviation (in hours)**	**Reduction (in %)**	**GAP (in %)**	**Running time**
1	112.8	1.3	99%	0%	57
2	101.0	1.2	99%	0%	62
3	161.9	14.8	91%	0%	616
4	157.4	10.1	94%	0%	1582
5	154.3	2.4	98%	4%	3601
6	157.1	4.0	97%	2%	3600
7	132.3	7.9	94%	0%	337
8	146.5	8.8	94%	2%	3601
9	146.4	1.8	99%	11%	3601
10	227.4	30.4	87%	7%	3600
11	136.1	10.1	93%	0%	1039
12	150.6	10.9	93%	0%	198
13	119.0	6.5	95%	0%	15
14	140.4	9.5	93%	0%	64
15	183.7	22.1	88%	0%	1522
16	143.2	8.0	94%	0%	290
17	155.9	10.9	93%	0%	397
18	167.2	5.3	97%	0%	206
19	151.4	4.3	97%	0%	347
20	144.4	13.9	90%	0%	378
21	136.8	4.6	97%	0%	425
22	129.0	5.3	96%	0%	784

23	137.3	7.8	94%	0%	50
24	172.1	10.3	94%	0%	208
25	156.1	12.4	92%	0%	495
26	147.3	15.3	90%	0%	63
27	147.7	13.2	91%	0%	255
28	109.0	6.8	94%	0%	463
29	108.5	7.3	93%	0%	73
30	124.5	1.4	99%	0%	83
31	115.2	6.0	95%	0%	157
32	132.5	8.2	94%	0%	370
33	122.3	7.1	94%	0%	92
34	132.8	8.4	94%	0%	2233
35	130.6	1.8	99%	0%	64
36	173.2	9.9	94%	0%	128
37	129.7	3.4	97%	0%	87
38	165.8	7.5	95%	0%	248
39	154.5	10.3	93%	0%	95
40	175.1	3.6	98%	0%	193
41	150.7	5.72	96%	0%	246
42	171.4	3.4	98%	0%	302
43	153.4	4.32	97%	0%	177
44	208.1	0.4	100%	0%	261
45	169.7	3.92	98%	0%	249
46	172.9	3.82	98%	0%	143
47	144.9	3.72	97%	0%	220
48	150.9	6.62	96%	0%	86
49	168.9	3.52	98%	0%	207
50	165.9	4.72	97%	0%	121
51	152.9	3.62	98%	0%	179
52	133.8	2.9	98%	0%	108
53	106.5	1.1	99%	0%	31
Average	**147.9**	**7.2**	**95.2%**	**0.5%**	**642**

Soft constraints

	Current situation	**Soft constraints**				
Week	**Total deviation (in hours)**	**Total deviation (in hours)**	**Penalty**	**Reduction (in %)**	**GAP (in %)**	**Running time**
1	112.8	1.1	0.1	99.0%	0.0%	27
2	101.0	1.2	0.0	98.8%	0.0%	37
3	161.9	11.7	2.9	92.8%	0.8%	3601
4	157.4	6.8	2.4	95.7%	0.0%	1256
5	154.3	2.4	0.0	98.4%	0.0%	401
6	157.1	3.7	0.2	97.6%	0.0%	913

7	132.3	4.8	2.4	96.4%	0.0%	3304
8	146.5	7.9	0.0	94.6%	0.0%	2225
9	146.4	1.4	0.2	99.0%	0.0%	938
10	227.4	29.1	2.2	87.2%	1.9%	3600
11	136.1	9.7	0.2	92.9%	0.0%	777
12	150.6	5.5	3.6	96.3%	0.0%	132
13	119.0	5.8	0.0	95.1%	0.0%	32
14	140.4	9.1	0.0	93.5%	0.0%	78
15	183.7	21.5	0.4	88.3%	0.5%	3600
16	143.2	4.0	0.5	97.2%	0.0%	205
17	155.9	6.4	3.6	95.9%	0.0%	470
18	167.2	5.3	0.0	96.8%	0.0%	177
19	151.4	4.3	0.0	97.2%	0.0%	189
20	144.4	9.6	3.5	93.3%	0.0%	646
21	136.8	4.6	0.0	96.6%	0.0%	523
22	129.0	5.3	0.0	95.9%	0.0%	241
23	137.3	7.9	0.0	94.2%	0.0%	127
24	172.1	10.3	0.0	94.0%	0.0%	242
25	156.1	12.0	0.0	92.3%	0.0%	712
26	147.3	14.9	0.0	89.9%	0.0%	100
27	147.7	8.9	3.5	94.0%	0.0%	320
28	109.0	6.8	0.0	93.8%	0.0%	648
29	108.5	7.3	0.0	93.3%	0.0%	114
30	124.5	1.4	0.0	98.9%	0.0%	104
31	115.2	6.0	0.0	94.8%	0.0%	273
32	132.5	8.2	0.0	93.8%	0.0%	255
33	122.3	7.1	0.0	94.2%	0.0%	82
34	132.8	8.4	0.0	93.7%	0.0%	319
35	130.6	1.9	0.0	98.5%	0.0%	83
36	173.2	9.6	0.0	94.5%	0.0%	114
37	129.7	3.4	0.0	97.4%	0.0%	99
38	165.8	7.5	0.0	95.5%	0.0%	178
39	154.5	10.3	0.0	93.3%	0.0%	104
40	175.1	3.6	0.0	97.9%	0.0%	121
41	150.7	5.7	0.0	96.2%	0.0%	157
42	171.4	3.4	0.0	98.0%	0.0%	149
43	153.4	4.3	0.0	97.2%	0.0%	167
44	208.1	0.4	0.0	99.8%	0.0%	494
45	169.7	3.9	0.0	97.7%	0.0%	294
46	172.9	3.8	0.0	97.8%	0.0%	368
47	144.9	3.7	0.0	97.4%	0.0%	153
48	150.9	6.6	0.0	95.6%	0.0%	151
49	168.9	3.5	0.0	97.9%	0.0%	150
50	165.9	4.7	0.0	97.2%	0.0%	177
51	152.9	3.6	0.0	97.6%	0.0%	175

52	133.8	2.9	0.0	97.8%	0.0%	117
53	106.5	1.1	0.0	99.0%	0.0%	71
Average	**147.9**	**6.5**	**0.5**	**95.7%**	**0.1%**	**566**