wirtschafts
wissenschaften
htw saar

A. Bernhardt I T. Melo I T. Bousonville I H. Kopfer

Scheduling of driver activities with multiple soft time windows considering European regulations on rest periods and breaks

Schriftenreihe Logistik der Fakultät für Wirtschaftswissenschaften der htw saar

Technical reports on Logistics of the Saarland Business School

Nr. 12 (2016)

logistik

ISSN 2193-7761

Scheduling of driver activities with multiple soft time windows considering European regulations on rest periods and breaks

A. Bernhardt I T. Melo I T. Bousonville I H. Kopfer

Bericht/Technical Report 12 (2016)

Die Veröffentlichungen in der Berichtsreihe der Fakultät für Wirtschaftswissenschaften können bezogen werden über:

Hochschule für Technik und Wirtschaft des Saarlandes
Fakultät für Wirtschaftswissenschaften
Campus Rotenbühl
Waldhausweg 14
D-66123 Saarbrücken

Telefon: +49 (0) 681/ 5867 - 519
E-Mail: fakultaet-wiwi@htwsaar.de
Internet: www.htwsaar.de/wiwi

Scheduling of driver activities with multiple soft time windows considering European regulations on rest periods and breaks

A. Bernhardt*[1], T. Melo[2], T. Bousonville[2], H. Kopfer[1]

[1]Chair of Logistics, University of Bremen, Bremen, Germany

[2]Institute for Supply Chain and Operations Management, Saarland University of Applied Sciences, Saarbrücken, Germany

November 16, 2016

Abstract

When considering long-haul transport requests, the durations of rest periods and breaks highly influence the overall time needed for fulfillment. In the European Union, Regulation (EC) No 561/2006 defines the rules for the number, duration and time intervals when rest periods and breaks have to be taken. The present study proposes two mixed integer linear programming models and optimization strategies that, together with a transformation algorithm, allow to plan driver activities in compliance with this regulation for a given sequence of customer locations and other stops to be visited. One of the models considers all rules, including extended rules, while the other takes into account the regular requirements. Each customer location has one or multiple time windows among which a choice has to be made. A special feature is the consideration of "soft" time windows which has not been studied in this context so far. If time windows cannot be met, the resulting schedule gives important information to the dispatcher that is necessary to set up a better schedule. In online re-planning, lateness can be revealed at an early stage such that it is possible to reorganize the schedule or to negotiate arrival times with customers before communication effort and costs increase and further delays or cancellations are unavoidable. In addition to the mathematical models, a myopic algorithm was developed that can only "see" the route until the next customer stop and the corresponding customer time window in advance and plans driver activities accordingly. Simple strategies were chosen to also integrate the optional rules. Test instances were derived from real data and include vehicle routes for one week. The numerical results obtained with the mathematical models and the myopic algorithm are analyzed and compared in terms of the run time, lateness and overall travel time.

Keywords: road transportation, driver scheduling, rest periods, breaks, driving hours, Regulation (EC) No 561/2006, mixed integer linear programming models

*Corresponding author. E-mail address: research@alexandra-bernhardt.net

Contents

List of Figures

List of Tables

1 Introduction

EU legislation aims at ensuring road safety, adequate working conditions and undistorted competition in the road haulage sector (European Commission (2014)). Regulation (EC) No 561/2006[1] regulates driving time and rest periods and Directive 2002/15/EC[2] the working time of drivers in road transport. While Regulation (EC) No 561/2006 on driving time and rest periods of drivers in road transport is itself obligatory in all member countries, for Directive 2002/15/EC additional national regulations also have to be taken into account.

To control the compliance with the above regulations also referred to as European social legislation, the European Parliament and Council of the European Union (2006a) determines the minimum level of enforcement. The rules provide for checks by authorized inspection officers in the range of 3 to 4 % of days worked by drivers. They determine the minimum proportion of checks on the roadside (30 %) and at the premises of undertakings (50 %). Road side checks shall be performed at any time and for example take place at service stations or at any other safe locations along motorways with the goal to cover the road network sufficiently. In addition to planned checks at premises in accordance with experience of the past, serious infringements detected are a reason for checks at premises of the corresponding undertakings. To expand and simplify checks, the introduction of the digital tachograph as recording equipment for driver activities was an important step. Its installation is obligatory in all new vehicles that have a mass of more than 3.5 tonnes since 2006. Its application is regulated by Regulation (EU) No 165/2014 (European Parliament and Council of the European Union (2014))[3].

Transport companies have to organize the work of drivers and instruct them such that they can comply with the social legislation. In particular, Regulation (EC) No 561/2006 stresses the responsibility of all members involved in the transportation process: "Undertakings, consignors, freight forwarders, tour operators, principal contractors, subcontractors and driver employment agencies shall ensure that contractually agreed transport time schedules respect this Regulation" (European Parliament and Council of the European Union (2006b)). For infringements, drivers as well as transport companies may be held responsible and fined severely - and not without reason.

According to the European Road Safety Observatory, in 2013, more than 26,000 people died on the roads of the European Union and more than 1.4 million people were injured. More than 15 % of people who died in road accidents in 2013 died in accidents that involved heavy goods vehicles with more than 3.5 tons maximum permissible gross weight (European Road Safety Observatory (2015)). Driver fatigue is involved in 10 % to 20 % of all road accidents and several studies suggest that it leads to an increased crash risk. Tired drivers tend to be unfocused, reaction times are increased and keeping distance to the vehicle in front can be difficult (SafetyNet (2009)). Sufficient rest periods and breaks help to keep the number of accidents as a consequence of driver fatigue low.

[1] amended by European Parliament, Council of the European Union (2009) and European Parliament and Council of the European Union (2014)

[2] European Parliament and Council of the European Union (2002) amended by European Commission (2009) and European Commission (2009)

[3] repealing Council Regulation (EEC) No 3821/8 (Council of the European Union (1985))

In the transport business, quality of service is an important key factor (European Commission (2014)) that includes punctuality at customer locations. In the EU, converging cost structures will more and more urge transport undertakings to "[...] improve their efficiency and quality of service" (European Commission (2014)). Lateness may be crucial, as it may cause contractual penalties and may lower customer satisfaction that has a significant impact on future requests and thus on the economic viability of a haulage company. The increase in just-in-time management practices even raises the pressure on truck drivers, dispatchers and their transport companies. Besides contractual penalties, a driver that arrives too late at a customer location and thus misses the planned time window may have to wait for several hours until loading or unloading is possible. Additionally, re-planning ties up resources at the transport company and the customer. An unnecessary early arrival time at a customer location is undesirable as well.

Considering long-haul transport requests, the durations of rest periods and breaks highly influence the overall time needed for fulfillment. They have to be considered when determining arrival times and planning time windows at customer locations and it is important to not only consider their duration but also their time slot in the schedule. For example, daily rest periods and breaks may be split in two parts, but these may not be further divided. At least nine uninterrupted hours are necessary for a daily rest period in which the driver is not allowed to drive or to perform other work. If the schedule only does consider resting activities in the form of a fixed proportion of the overall traveling time, deviations from planned arrival times will occur frequently as rest periods and breaks can not be interrupted or split arbitrarily to serve customers and this will lead to the problems described above, especially if narrow time windows are involved.

2 Motivation

In commercial road transport, typically dispatchers are responsible for planning vehicle routes, instructing the drivers with respect to the sequence in which customer locations should be visited and to plan arrival times at customer locations. To avoid lateness is essential in this context.

When planning arrival times of long-haul trips that require several days, Regulation (EC) No 561/2006 concerning driving and working hours of drivers in road transport is obligatory in all member countries of the EC. The regulation aims at improving safety and working conditions of drivers in road transport and has a high influence on the execution time of a transport request. Disregarding corresponding rules may be fined severely. So, despite the rules being rather complex in their application as often many different possibilities to plan driver activities have to be evaluated, a dispatcher has to set up his plans assuring that drivers are able to strictly comply with the regulation.

Today, vehicle telematics can be used to provide dispatchers with up-to-date information on current vehicle positions and often give further information, for example, on the driver status concerning driving times, rest periods and breaks, but still the problem of planning arrival times remains non-trivial to be solved. If the consideration of rest periods and breaks is not integrated in a direct way into the planning process, the dispatcher may choose to use rough time buffers when estimating possible arrival times and negotiating time windows with customers. In many cases though it may be the case that time buffers have been chosen too large or too short which will result in lateness and/or unnecessary waiting times. On the one hand, repeated lateness may lead to a loss of customer requests. On the other hand, the execution time may be prolonged artificially by unnecessary waiting times which may result in more costs for the freight forwarding company.

Thus, it seems worthwhile to precisely schedule driver activities when planning arrival times at customer locations. As there are plenty of information and possibilities to evaluate, models and algorithms considering restrictions on working hours that are embedded in decision support and planning tools can be an important value added to help dispatchers to plan vehicle movements and to estimate arrival times at customer locations.

We study a problem motivated by the truckload shipping services offered by a medium sized company operating in Europe. A significant part of the shipping orders of the company result from fixed contracts, whereas several partner companies pass on requests as well. Additionally, requests are acquired at freight exchanges to supplement partial loads and to use the load capacity of the vehicles as efficiently as possible. Vice versa, shipping orders may be passed on to subcontractors as well. An on-site process analysis gave detailed insights into the planning of vehicle routes and the distribution of responsibilities among dispatchers and drivers. Dispatchers decide about transport requests and plan vehicle routes, arrival times at customer locations and time windows. Each request for transportation consists of a pickup and a corresponding delivery location. These locations are often far apart and only a few customers can be serviced during the same week. Often, the fulfillment of a transport request extends to the following week. Then, the driver does not return home but takes his weekly rest period somewhere near to the route. Many drivers return home only after several weeks.

A feature observed and that is common in long-haul freight transportation is that customer orders are placed with several alternative time windows. Hence, the construction of routes also involves the choice of time windows.

In a manual planning phase, the dispatchers assign transportation requests to vehicles and determine the sequence in which customer locations shall be visited. Only a part of the transportation requests are known at the start of the week. As the time unfolds, additional requests are accepted and assigned to vehicles. The estimated duration of travel between locations includes a time buffer for rest periods and breaks that is proportional to the distance traveled. Arrival times at customer locations are planned accordingly and drivers are informed. Drivers are responsible to plan rest periods and breaks and may negotiate arrival times with the dispatcher. Several large customers propose time windows for loading or unloading among which dispatchers have to choose. Favorable time windows may be occupied if not chosen early enough but a reliable estimation of arrival time is necessary. Therefore, exact time windows are not necessarily scheduled when accepting transport requests, they are chosen when the arrival time at the customer location is predictable, i.e. one or two days in advance depending on the demand.

3 Objective

The initial situation is described by a sequence of stops that are planned to be visited during the current week by a vehicle, each having one or multiple alternative time windows.[1]

Distances between consecutive stops and estimated durations for loading, unloading and handling activities are given. Planning may start at the beginning of the week but is also possible during the week. The time management of a driver since the end of the last weekly rest period influences his future activities and is therefore accumulated to the driver status and serves as input for the current (re-)planning phase. This also allows the consideration of online re-planning, i.e adjustments of the original schedule to dynamically respond to unforeseen events.

The goal is to construct a driver schedule that simultaneously considers the choice among possible customer time windows and plans necessary rest periods and breaks to minimize inefficiencies that arise from the distributed decision making of drivers and dispatchers to increase punctuality and to avoid unnecessary time buffers. In some particular cases, it may not be possible to meet any of the available time windows at a given location. As the dispatcher often has the possibility to negotiate the arrival time of the vehicle with the customer at an early stage, we consider soft time windows that can be violated at a penalty. Another reason for taking soft time windows is that we want to give detailed information even if time windows cannot be met to help dispatchers in the re-planning phase.

For this task, we developed two mixed integer linear programming (MILP) models considering a maximum planning horizon of one week. In one of the MILP models, the optional rules of Regulation (EC) No 561/2006 are taken into account, in the other one, the optional rules are disabled. The main objective of the two multiciteria optimization problems considered is to minimize the sum of the overall lateness. With less emphasis the completion time, i.e. the overall schedule duration until the last customer is serviced and the last stop is reached[2] is the second optimization criterion. These two criteria form the first set of optimization criteria. Other criteria that are important for the quality of a solution in practice are taken into account in a second set. As the first set (i.e. lateness and completion time) is considered to be more important than the second one, a lexicographic solution strategy was chosen by creating two objective functions, the first one for the first set, the second one for the second set. Using a trade-off strategy within each of the two sets, in each of the objective functions the criteria were provided with different weights. Test instances were derived from real life data. The two MILP models were solved to optimality using a state-of-the-art commercial solver and driver schedules were created from optimal model solutions with the help of a transformation algorithm. Test results were analyzed.

The effort to integrate an optimization solver must be worthwhile. A myopic algorithm was also implemented that runs without such software as an alternative in order to identify advantages and disadvantages of the MILP models and a solver.

[1] If a stop corresponds to a customer location with given opening hours, then time windows covering the opening hours reflect this situation.

[2] Stops that do not correspond to a customer location are possible. For example, the last stop may be a depot or a rest area that is chosen for the weekly rest period.

The following sections are structured as follows. In Section 4 the rules implied by regulation Regulation (EC) No 561/2006 are described. Section 5 gives a short review of literature that deals with scheduling driver activities in compliance with Regulation (EC) No 561/2006. In Section 6 the MILP models with and without consideration of the optional rules are introduced, thereby a short digression on modeling techniques used is presented. Afterwards, in Section 7 the derivation of test instances is described and different approaches for solving the MILP model with consideration of the optional rules are discussed. Run times for a set of instances based on real-life data are analyzed depending on the number of stops and the number and length of time windows. To examine the influence of the optional rules, the two MILP models are compared with each other considering run times, lateness and completion time. Then, the myopic algorithm is presented in Section 8. With the test instances presented before, tests are repeated with the short-sighted algorithm and a comparison to the former test results is made in Section 9. Section 10 gives a short summary and suggestions for future research.

4 Rules

Regulation (EC) No 561/2006 aims at improving working conditions and safety of drivers in road transport laying down provisions concerning maximum driving periods and necessary breaks and rest periods. It applies to the carriage of goods where the maximum permissible mass of the vehicle exceeds 3.5 tonnes or of passengers by vehicles that may carry more than nine persons. It affects all transports exclusively within the European Community or between the European Community, Switzerland and the countries that are part of the Agreement on the European Economic Area. The regulation comprises rules for single drivers and multi-manning.

For all possible driver activities, Regulation (EC) No 561/2006 defines several time periods with specific rules (see Meyer and Kopfer (2008) and Figure 1). The rules can be divided in standard rules and optional rules, where adhering to the standard rules suffices to observe the law, while the optional rules allow for more freedom providing alternatives for some of the standard rules.

Before introducing the rules, we give some basic definitions for time periods.

Definitions:

- A *rest period* is any uninterrupted period of time during which a driver may freely dispose of his or her time. *Daily rest periods* and *weekly rest periods* are rest periods.
- A *break* is a time period exclusively designed for recuperation, during which a driver may not carry out any driving or any other work. Other work comprises various activities such as loading or unloading, cleaning, technical maintenance, administrative formalities, ensuring safety of the vehicle and its load, etc. Waiting times that are not known in advance are also considered as other work, as the driver cannot dispose freely of his time and is required to be at his workstation. A break has a duration of at least 45 minutes.

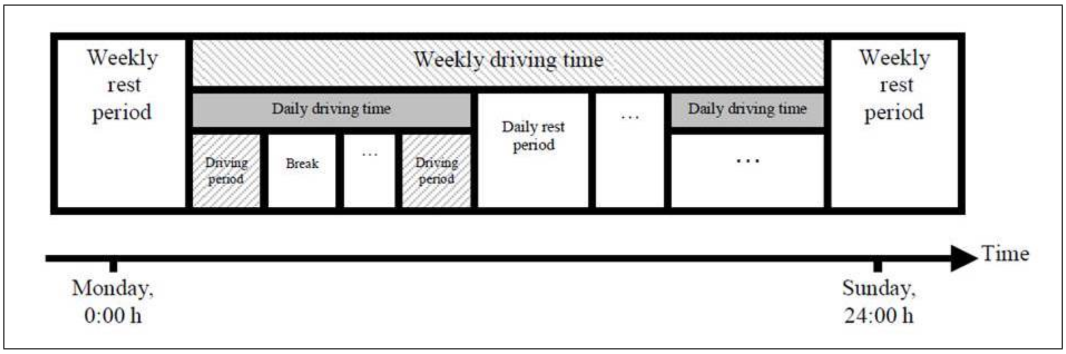

Figure 1: Relation of the different time horizons (Meyer and Kopfer (2008))

- *Driving time* is the duration of driving and includes all activities related to driving, even when the vehicle is temporarily not in motion, for example, when waiting at traffic lights or in a traffic jam.
- A *week* means the period of time between 00:00 on Monday and 24:00 on Sunday.

Standard rules:

1. A break has a duration of at least 45 minutes.
2. A daily rest period has a duration of at least 11 hours.
3. A weekly rest period has a duration of at least 45 hours.
4. The accumulated driving time between a rest period or a break and another rest period or break is restricted to a maximum of 4.5 hours.
5. The daily driving time, i.e. the total accumulated driving time between the end of one rest period and the beginning of the following rest period, is restricted to a maximum of 9 hours.
6. Within each period of 24 hours after the end of the previous rest period a driver must have taken a new daily rest period. This means that a driver must take a daily rest period at most 13 hours after he has completed the previous daily or weekly rest period.
7. A weekly rest period shall start no later than 144 hours (six 24-hour periods) after the end of the previous weekly rest period.
8. The *weekly driving time*, i.e. the total accumulated driving time between 00:00 on Monday and 24:00 on Sunday, is not allowed to exceed 56 hours.
9. The total accumulated driving time during any two consecutive weeks must not exceed 90 hours.
10. In any two consecutive weeks a driver has to take at least two weekly rest periods. Weekly rest periods that fall in two weeks may be counted in either week, but not in both.

11. The maximum weekly working time is not allowed to exceed 60 hours.
12. Over four months, the average weekly working time should not exceed 48 hours.

The last two regulations 11 and 12 on the weekly working time stem from Directive 2002/15/EC (European Parliament and Council of the European Union (2002)), as Regulation (EC) No 561/2006 refers to Directive 2002/15/EC for laying down the maximum weekly working time.

Optional rules:

1. A break may be taken in two parts, the first part having a duration of at least 15 minutes followed by the second one of at least 30 minutes.
2. A regular daily rest period may also be split into two parts with the first one having a duration of at least 3 hours and the second one having a duration of at least 9 hours.
3. The duration of a daily rest period may be reduced to at least 9 hours at most 3 times between two weekly rest periods. In the case that the next daily rest period is planned to be a reduced one, that rest period has to start at most 15 hours after the completion of the previous daily or weekly rest period.
4. The daily driving time may be extended to at most 10 hours not more than twice during a week, where a week means the period of time between 00:00 on Monday and 24:00 on Sunday.
5. In two consecutive weeks, the duration of one of the two weekly rest periods may be reduced to 24 hours. However, the reduction has to be compensated by an equivalent period of rest taken at a time before the end of the third week following the week containing the reduced weekly rest period. Any rest taken as compensation for a reduced weekly rest period should be attached to another rest period of at least 9 hours.

Deviating from the standard rules 2 and 6 and making the optional rules 2 and 3 needless, a driver engaged in multi-manning must have taken a new daily rest period with a duration of at least 9 hours within 30 hours of the end of a daily or weekly rest period. In the following, we will focus on the single truck driver case.

Furthermore, additional rules apply for special cases as for example for accompanying vehicles transported by ferry or train, the traveling to a location to take charge of a vehicle, for driving times not falling in the scope of Regulation (EC) No 561/2006 or special vehicles.

For transport between EU and non-EU countries (third countries) the AETR agreement (see United Nations Economic Commission for Europe (2006)) regulates the work of drivers engaged in international transport and covers 49 contracting parties including all EU Member States (European Commission (2016)). The implementation into national law is mandatory. Its provisions concerning rest periods and breaks are similar to Regulation (EC) No 561/2006, with the difference that drivers engaged in multi-manning are exempted from standard rule 10 and optional rule 5.

5 Literature review

In the past few years, research on including regulations concerning rest periods and breaks in operational transportation planning has attracted increasing attention especially in combination with vehicle routing. The following review concentrates on literature that deals with Regulation (EC) No 561/2006 and the single truck driver case.

Zäpfel and Bögl (2008) and Bartodziej et al. (2009) consider some of the rules implied by Regulation (EC) No 561/2006 in their rich vehicle routing problems set up in real case studies.

Goel and Gruhn (2006) and Goel (2009) present a large neighborhood search algorithm to solve the combined problem of vehicle routing with time windows and scheduling driver activities in consideration of Regulation (EC) No 561/2006 for single manned vehicles and for a planning horizon of one week. For the scheduling sub-problem Goel (2009) introduces a naive labeling algorithm and a multilabeling algorithm to plan driver activities according to a subset of the regulations. The naive method only schedules breaks and rest periods if the accumulated driving time is exhausted or enough idle time before the lower boundary of the time window at a customer is available, whereas the multilabeling method allows for earlier breaks and rest periods. It is shown that in some cases it may be beneficial to schedule breaks or rest periods before the corresponding accumulated driving time reaches its maximum to be able to reach narrow time windows. Optional regulations are excluded and the regulation that there has to be a daily rest period within 24 hours after the end of the previous daily rest period (standard rule 6) is only considered in a post processing step by Goel (2009), who presents a repair method. New benchmark instances based on the well-known instances by Solomon (1987) for the vehicle routing problem with time windows (VRPTW) are set up.

Derigs et al. (2011) extend the work of Bartodziej et al. (2009) and present a checking procedure for route feasibility that is motivated by Goel (2009). Splitting of daily rest periods and breaks, reduced daily rest periods and the possibility to extend the daily driving time are considered in the procedure.

In addition to Goel (2009), Kok et al. (2010) consider all optional rules and also Directive 2002/15/EC that supplements the rules laid down in Regulation (EC) No 561/2006 with additional restrictions for the working time of persons engaged in road transportation, extending the naive labeling method presented by Goel (2009). They incorporate the EC social legislation in a restricted dynamic programming algorithm by adding state dimensions. Breaks are scheduled in constant time by using a constructive solution method with a break scheduling algorithm that decides locally when breaks have to be scheduled. Their test results show significant improvements concerning the number of vehicles needed and the distance traveled even though they allow less computation time than Goel (2009). In particular, Kok et al. (2010) show that including the optional rules of Regulation (EC) No 561/2006 allows for additional flexibility and can reduce costs significantly. However, similar to Goel (2009), they cannot guarantee to find a feasible driver schedule for a route, even when one exists.

Given a sequence of locations to be visited within specified time windows, Goel (2010) presents a method for scheduling driving and working hours of truck drivers with respect to Regulation (EC) No 561/2006. The feature of this approach is the guarantee to find a schedule complying with the regulation if such a schedule exists. Goel (2010) introduces conditions for pseudo-feasibility which relax the conditions for feasibility and gives dominance criteria, thus reducing the number of partial schedules that have to be explored. However, he neglects the possibility of extending daily driving times and of reducing the duration of daily rest periods.

Drexl and Prescott-Gagnon (2010) describe an exact and two heuristic approaches for considering European rules for rest periods and breaks in shortest path problems with resource constraints. The labeling algorithms are based on the idea of so-called resource extension functions to expand labels to plan rest periods and breaks.

Prescott-Gagnon et al. (2010) present a large neighborhood search method for the vehicle routing problem with time windows and driver regulations. In this method, the neighborhoods are explored using a column generation heuristic that relies on a tabu search algorithm. This tabu search algorithm allows two possible route modifications, the deletion or the insertion of a customer. While route feasibility is maintained if a customer is removed, the insertion of a new customer requires a feasibility check. Therefore, Prescott-Gagnon et al. (2010) develop a heuristic procedure that uses labels with resource components and resource extension functions. Numerical results show that significant improvements can be achieved compared to the procedures proposed by Kok et al. (2010) and Goel (2009).

Kopfer et al. (2007) analyze the influence of the European social legislation on vehicle routing and scheduling and are the first ones to consider a modeling approach. Later, Kopfer and Meyer (2009), Kopfer and Meyer (2010), and Meyer (2011) develop a MILP formulation to map Regulation (EC) No 561/2006. Kopfer and Meyer (2009) use a position-based formulation of the traveling salesman problem with time windows (TSPTW) to integrate the rules of the regulation. Kopfer and Meyer (2010) and Meyer (2011) continue with an extension for the VRPTW, also including Directive 2002/15/EC. They solve randomly generated test instances with CPLEX. Unfortunately, if long distances have to be traveled between two consecutive customers, the model is not applicable, as it is presumed that driving between two customer locations will not require more than one daily rest period. The constraints that are used to model the restriction that there has to be a daily rest period in each 24 hours time interval only demand a number of daily rest periods proportional to the overall travel duration and thus are problematic if daily rest periods are scheduled earlier than required. The solution specifies the number of breaks and rest periods between two consecutive customers. The transformation algorithm that is necessary to determine a driver schedule that includes the timing and the sequencing of rest periods and breaks is not described in detail.

Kok et al. (2011b) present a model for departure time optimization as a post-processing step of the VRPTW that incorporates the European driving hours regulations and time-dependent travel times. They assume that breaks have to be taken at customer locations. Here, a planning horizon of a working day is considered and computational results are discussed. Additional constraints to model a planning horizon of multiple days and the possibility to take breaks at parking lots are proposed, but the option of splitting daily rest periods into two parts is neglected. No test results for the extended model are presented.

Kok et al. (2011a) propose a restricted dynamic programming heuristic for the VRPTW with time-dependent travel times and EC social legislation that is restricted to a planning horizon of one day.

Goel (2012) presents a mixed integer programming formulation for a variant of the truck driver scheduling problem in which drivers only may rest at customer locations and rest areas and shows how to model rules commonly found in different hours of service regulations. Rest areas are modeled as dummy locations with zero duration for loading and unloading and unbounded time windows. A dynamic programming approach is proposed that is able to solve the problem efficiently and it is shown how additional rules like the optional rules of Regulation (EC) No 561/2006 for splitting breaks and daily rest periods can be incorporated. However, it is assumed that rest areas are roadside, as no detours are considered. Test instances are randomly generated for a planning horizon of one work week that ends on Friday, rest areas are randomly distributed, and up to 4 time windows per customer location with time windows from 6:00 to 20:00 or from 6:00 to 12:00 and 14:00 to 20:00 on one or two days are considered. It is shown that the availability of suitable rest areas has a significant impact on the number of instances for which feasible schedules could be found.

Goel and Vidal (2014) use a hybrid genetic search with advanced diversity control for solving the combined vehicle routing and truck driver scheduling problem. Truck driver scheduling is done for route evaluations with adjustments of the forward labeling algorithms developed for the rules applied in different countries and areas, among these the rules of the European Union. Considering Directive 2002/15/EC, the authors include the same set of rules as Prescott-Gagnon et al. (2010). They consider multiple time windows and allow penalized lateness with respect to the time window constraints. However, lateness is only allowed to facilitate transition between structurally different solutions during the search and there, any voluntary increase in lateness at a customer location for the purpose of reducing lateness at subsequent customers is forbidden. Furthermore, Goel and Vidal (2014) give an international comparison of the economic impact of different hours of service regulations.

6 Mathematical formulation

We start by describing the initial situation. A dispatcher has assigned transportation requests to a vehicle. Each request consists of a pickup and a corresponding delivery location (customer locations). The dispatcher has determined a sequence in which the customer locations have to be visited (see for example Figure 2).

Each customer location has at least one time window, i.e. a time interval in which the loading and/or unloading of goods should start. In contrast to Kopfer and Meyer (2010), we consider multiple customer time windows, as in reality, a dispatcher often has the possibility to choose among a set of time windows proposed by a customer. Loading or unloading does not have to be finished before the end of the chosen time window. This is just a modeling decision. If in practice the loading and unloading should take place within the time window, the estimated duration is subtracted from the end of the interval

Figure 2: Example of a sequence of vehicle stops over a time horizon of one week

to narrow the time window accordingly. Note that a time window that lies beyond the maximum time interval of $6 \cdot 24\,\text{h} = 144\,\text{h}$, that starts with the end of the last weekly rest period, will avoid finding a feasible schedule, as standard rule 7 cannot be met. This can be checked in advance.

Modeling time windows as hard constraints like it is done in the literature (recall Section 5) has one major drawback. If no schedule can be found because time windows cannot be met, the dispatcher only gets the information that there is "no solution". We use soft time windows, that means that we penalize lateness but do not prohibit it (see objective function 6.184 and constraints 6.77) which has the advantage that still a schedule may be returned even if time windows cannot be met. Unavoidable lateness is revealed and the dispatcher is given a schedule that may help him to re-plan or re-negotiate time windows with the customers.

Only the constraints that take care of the maximum time interval between two consecutive weekly rest periods may avoid that a schedule can be found (see constraints 6.78 on page 43). In such a case, constraints 6.78 can be removed and the MILP model can be re-solved to obtain a solution that is not feasible in practice, but may be an important information for the dispatcher.

The current vehicle position or starting point of the tour and all customer locations are represented by vertices, where vertex 0 denotes the vehicle position at the start of the planning horizon (depicted as a green point in Figure 2) and vertices $1, \dots, r-2$ denote the customer locations (customer vertices) that are numbered in the order they have to be visited (see Figure 3). As drivers do not necessarily return to a specific location like a central depot when doing long-haul trips, the last vertex represents the last known customer location to be visited or, if known in advance, the location where the driver will take the weekly rest period (red point in Figure 2). That may be a rest area near to the route to the subsequent customer location or near to a customer location, if loading or unloading should start after the weekly rest period.

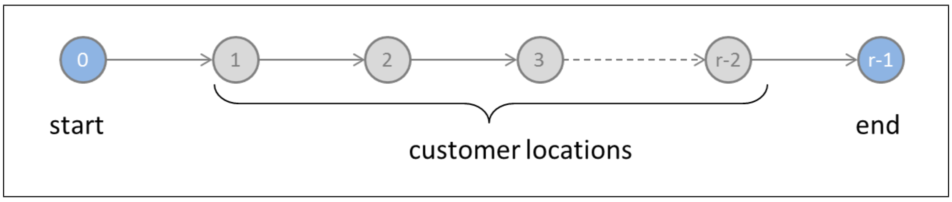

Figure 3: Vertices

In each customer vertex[1], the driver may wait, take a break or a daily rest period before loading or unloading. If the daily working time still left when entering a vertex does not suffice to also carry out the loading/unloading, a daily rest is scheduled. An interruption by daily rests is thus avoided. Generally, rest periods and breaks may be taken to reduce or

[1] For ease of reference, and to connect modeling aspects to the real life application, the term "vertex" is used in association with customer locations. Analogously, the term "arc" is used, when talking about the route between customer locations and activities on this route.

eliminate the waiting time. This also includes a first part of a splitted break or rest if this helps to reduce the time needed for resting activities on subsequent arcs and in vertices. Figure 4 summarizes the activities that can be performed at a vertex.

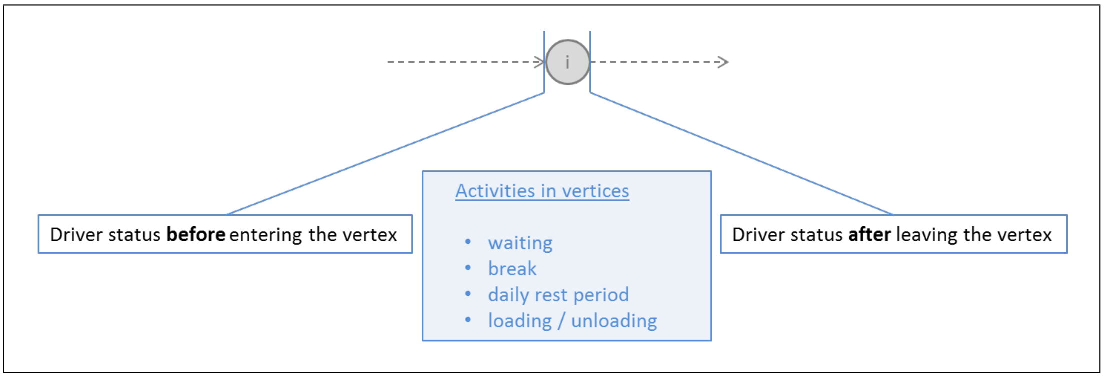

Figure 4: Driver time management activities in a vertex

Possible activities on an arc comprise driving, taking breaks and daily rest periods (see Figure 5). Note that the first part of a split break may only be scheduled in vertices to compensate waiting time, whereas the first part of a daily rest period may also be scheduled as last resting activity on an arc instead of a break. The first part of a daily rest period similar to a break resets the 4.5 hours driving contingent. If a first part of a daily rest period is taken, an additional break is only necessary if a driving time extension is considered. In some cases, it may be advantageous to take a first partial daily rest period instead of a break as last resting activity on an arc (for more details see Section 6.4.2 et seq.). Reduced daily rest periods and breaks needed for a driving time extension may be scheduled on arcs and vertices as well.

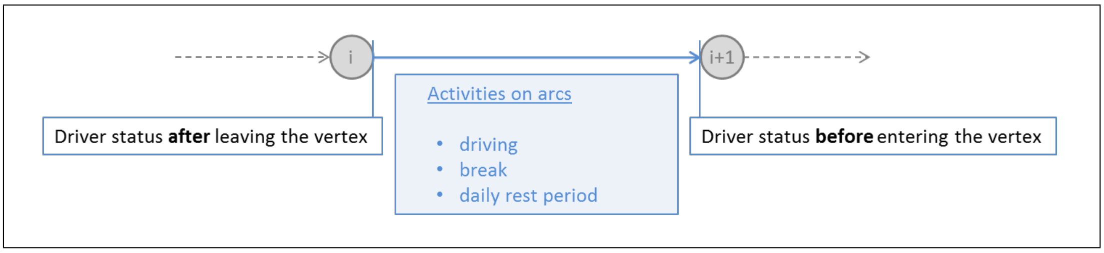

Figure 5: Driver time management activities on an arc

Similar to the literature presented in Section 5, we consider a planning horizon of at most one week that ends with the start of the next weekly rest period and concentrate on single manned vehicles. As the overall driving and working time needed is already given with the input data, a check can be made if the weekly driving (see standard rule 8) and working time (see standard rule 11) allowed suffice to fulfill all scheduled customer requests.[1] The

[1] Note that waiting time known in advance is not considered as working time.

biweekly working time (standard rule 9) can be considered by adding the working time of the last week to the planned working time for the current week and testing whether the result is less than 90 hours. Similarly, compliance with the average weekly working time over four months (standard rule 12) can be checked. Standard rule 10 and optional rule 5 are not considered by the MILP models and compliance with them has to be ensured externally when concatenating two planning horizons with the first one ending before the start of a weekly rest period and the other one starting after the end of the weekly rest period.

Usually, depending on the countries visited, it is advantageous to include Sundays in weekly rest periods because of the ban on movement of goods vehicles on Sundays in many countries of the European Union. Therefore, we assume that the time between two weekly rest periods does not range across more than one week, i.e. the time between 00:00 on Monday and 24:00 on Saturday.

Drivers may help when loading and unloading the vehicle but this does not have to be the case. At least, they have to perform handling activities. We assume that the drivers have to be present when loading or unloading takes place and consequently do not interpret this time as a break and do not allow it to be used as a part of a daily rest period.

In our models, no use is made of constraints which give a lower bound on the number of rest periods and breaks needed for every number of consecutive arcs as it is done in Kopfer et al. (2007). Instead, status variables are introduced which map the driver status when entering and leaving a vertex and thus link activities on different arcs and in different vertices. These status variables reflect, for example, the amount of time left for different activities without taking a break or daily rest when entering or leaving a vertex:

- E_i^{dt}, L_i^{dt}: driving time left until the next break or daily rest when entering vertex i or leaving, respectively.
- E_i^{ddt}, L_i^{ddt}: daily driving time left until the next daily rest when entering vertex i or leaving, respectively.
- E_i^{t}, L_i^{t}: overall time left until the next daily rest when entering vertex i or leaving, respectively.

Other status variables keep track of driving time extensions and reduced daily rest periods that were previously scheduled. Moreover, if a break or a daily rest period is split into two parts, status variables keep track if a first part has already been taken.

Dependencies between activities on arcs and the status variables are shown in Figure 5. Figure 4 depicts the dependencies when entering and leaving a vertex.

The models allow setting start values for the status variables. Thus, online re-planning during the week with updated tour information including remaining stops of the original schedule and new ones can take place. If there are deviations from the original plan, e.g. an increased travel time due to traffic congestion, and time windows cannot be met or the time left does not suffice to visit all locations planned, with online re-planning dispatchers get the possibility to recognize such deviations early and to re-negotiate time windows or remove stops from the vehicle schedule.

The models explore the possibility of taking a break or a daily rest period earlier than after 4.5 h of driving. This is advantageous if the time for a break or a rest period in a subsequent node or arc can be saved thus preventing a time window from being violated. This is illustrated in Figure 7. For an explanation of the symbols used, see Figure 6. The first schedule leads to a lateness of 15 minutes, whereas in the second schedule with an early daily rest the time window at customer $i+1$ is met.

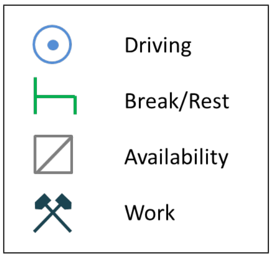

Figure 6: Symbols

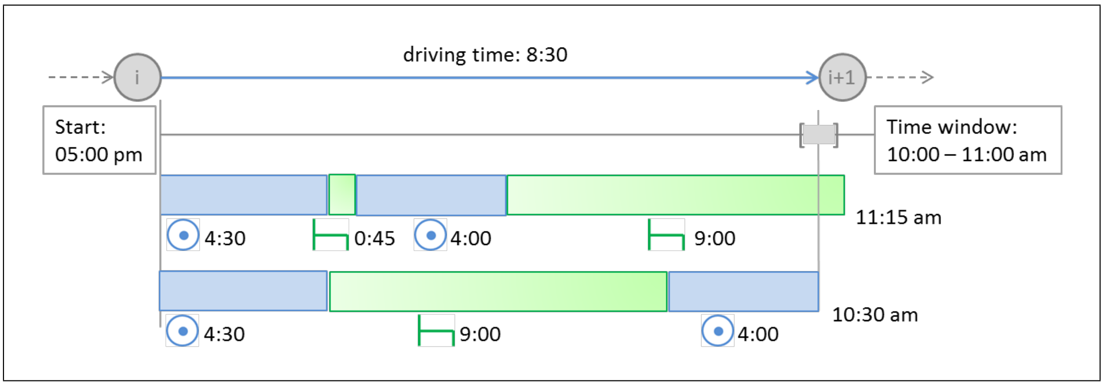

Figure 7: With vs. without early daily rest

Figure 8 shows an example, where taking a break in a vertex is necessary to be on time at the subsequent customer location.

With regard to an application in international freight transportation, where often long distances have to be traveled, we wish to exactly determine the time intervals for each planned driver activity. Therefore, an algorithm was developed to transform the model solution into a detailed driver schedule that gives information to both, the driver and the dispatcher. This algorithm can be found in the appendices (see appendix A).

The remainder of this section is structured as follows. After a short digression on modeling techniques, model parameters and variables will be introduced. Then, the MILP model with consideration of the optional rules is presented, whereby the description of the objective functions and the constraints is split into several subsections. How to switch off the optional rules is described at the end of this section. Note that all durations and lengths of time intervals are expressed in minutes.

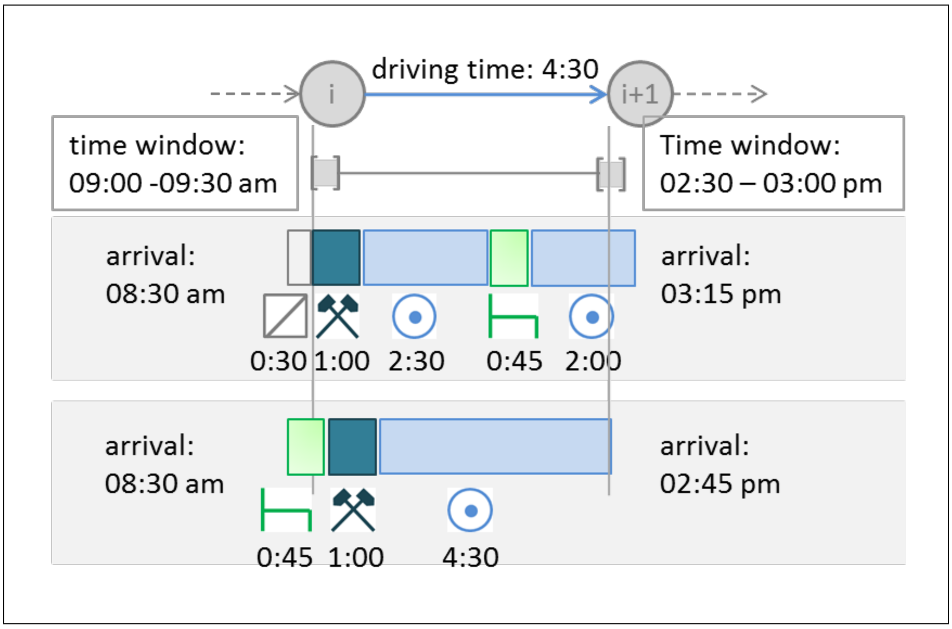

Figure 8: With vs. without a break in vertex i

6.1 Modeling techniques - A digression on modeling logical conditions with binary variables

For modeling different kinds of decisions such as taking a daily rest period or making use of one of the optional rules, and modeling different kinds of driver states that will be introduced later, binary (zero-one) decision variables will be used. For example, α_i^{rest} is a binary variable that takes the value 1 if a daily rest period should be taken at customer location i, and 0 otherwise.

As described in the previous section, allowed driver activities in a vertex depend on former decisions about activities on the previous arc. The latter in turn depend on activities scheduled for the previous vertex, and so on. To express these dependencies in the model, status variables link the activities in successive arcs and vertices. Activities are modeled by indicator and integer variables, whereas continuous variables reflect their duration. The goal here is to link the continuous status variables to indicator variables for activities.

Depending on the relation of several driver status variables to other driver status variables for entering or leaving a vertex i, the values of driver status variables for leaving vertex i or entering vertex $i+1$, respectively are determined. A good example is the daily driving time left L_i^{ddt} when leaving vertex i (see pages 61 et seq.). Here, indicator variables are used to determine if other status variables set up lower and upper bounds on L_i^{ddt}.

Generally, when formulating conditions, we can differentiate between the following two possibilities that reflect the direction of dependency:

- the value of a variable (continuous or integer) is derived from the value of one or several binary variables.
- the value of a binary variable is derived from other variables (binary, integer or continuous variables) and their relation to each other.

We will now go into further detail for the possibilities mentioned above. A more general description of transforming logical conditions into linear constraints by the usage of binary variables can be found in Williams (2013).

6.1.1 Binary variables inducing the value of other variables

We start by addressing the simple case that from the binary variable δ being equal to 1 it follows that the variable x is equal to y. y itself may be a linear expression. That means, we wish to state that

$$\delta = 1 \Rightarrow x = y \tag{a}$$

Therefore, we first reformulate (a) by (b):

$$\begin{aligned} &\delta = 1 \Rightarrow x \leq y \quad \text{and} \\ &\delta = 1 \Rightarrow x \geq y \end{aligned} \tag{b}$$

Now, the condition can be induced by two linear constraints:

$$x \leq y + \mathrm{M}_1\ (1 - \delta) \tag{c}$$
$$x \geq y - \mathrm{M}_2\ (1 - \delta) \tag{d}$$

M_1 has to be chosen as an upper bound on $x-y$ such that constraint (c) becomes redundant in case that $\delta = 0$. Similarly, M_2 has to be chosen as a lower bound on $x - y$. It is important not to choose M_1 and M_2 too small as this will introduce constraints on the difference between x and y that we do not want to model. In turn, taking very large values for M_1 and M_2 may result in numerical problems. Therefore, Williams (2013) recommends to choose upper bounds as small as possible and lower bounds as large as possible.

If more than one binary variable needs to take on 0 or 1 in order that x equals y, several "big-M terms" have to be added. As an example, we address the combination of two binary decision variables.

First of all, let us consider the case that the given values of two variables (example: $\delta_1 = 1$ and $\delta_2 = 0$) induce that $x = y$. The logical conditions

$$(\delta_1 = 1 \wedge \delta_2 = 0) \Rightarrow x = y \tag{e}$$

can be split into

$$(\delta_1 = 1 \wedge \delta_2 = 0) \Rightarrow x \leq y \tag{e1}$$
$$(\delta_1 = 1 \wedge \delta_2 = 0) \Rightarrow x \geq y \tag{e2}$$

This now can be modeled by the following linear constraints

$$x \le y + \mathrm{M}_1\ (1-\delta_1) + \mathrm{M}_1\ \delta_2 \tag{e3}$$
$$x \ge y - \mathrm{M}_2\ (1-\delta_1) - \mathrm{M}_2\ \delta_2 \tag{e4}$$

with M_1 and M_2 being appropriate upper and lower bounds, respectively.

If y itself is already an upper bound or lower bound on x, no additional constraint with a big-M term needs to be added to the model. If additionally to (e), it is known that already $\delta_1 = 1$ induces $x \le y$, condition (e3) is substituted by

$$x \le y + \mathrm{M}_1\ (1-\delta_1) \tag{e3'}$$

Analogously, this can be done if $\delta_1 = 1$ induces $x \ge y$ or if $\delta_2 = 0$ induces $x \le y$ or $x \ge y$.

The "exclusive or" in $(\delta_1 = 1\ \dot\vee\ \delta_2 = 1) \Rightarrow x = y$ is equal to the combination of

$$(\delta_1 = 1\ \wedge\ \delta_2 = 0) \Rightarrow x = y \quad \text{and}$$
$$(\delta_1 = 0\ \wedge\ \delta_2 = 1) \Rightarrow x = y$$

and can therefore be expressed by using the scheme described above. In case that

$$\delta_1 + \delta_2 \le 1$$

the following two additional inequalities suffice.

$$x \le y + \mathrm{M}_1\ (1-\delta_1-\delta_2)$$
$$x \ge y - \mathrm{M}_2\ (1-\delta_1-\delta_2)$$

This will be used when deriving the daily driving time left L_i^{ddt} when leaving vertex i on pages 61 et seq.

6.1.2 Binary variables derived from other variables

Now, the opposite direction is considered. The value of binary variables is determined depending on the value of other variables being either integer or continuous. A dependency we will often need later (see for example pages 61 for λ_i^1 to λ_i^3) is the following one: in case x is greater than y, the binary variable δ should be equal to 1 if x is less than y, δ is set to be equal to 0. If $x = y$, we do not care about the value of δ.[1]

[1] In principle, in the cases we have to consider, we have to model a piecewise linear continuous function depending on x. In the simplest case, this function f is composed of two parts with

$$f(x) = \begin{cases} g(x) \text{ if } x \le y \\ h(x) \text{ if } x > y \end{cases}$$

For $x = y$ it does not matter which function is evaluated (g or h) to determine $f(x)$, as $g(y) = h(y)$, because f is continuous.

This means that we wish to state that

$$x > y \Rightarrow \delta = 1$$
$$x < y \Rightarrow \delta = 0$$

This can be done by using the following constraints:

$$\mathrm{M}_1\ \delta \geq x - y \tag{f1}$$
$$\mathrm{M}_2\ (\delta - 1) \leq x - y \tag{f2}$$

where (f1) induces $x > y \Rightarrow \delta = 1$ and (f2) induces $x < y \Rightarrow \delta = 0$. M_1 has to be an upper bound on $x - y$ whereas $-\mathrm{M}_2$ has to be a lower bound on $x - y$. This procedure is used for determining the value of the binary variables λ_i^k, $k = 1, \ldots, 6$ which depend on several continuous status variables and activities in vertex i and arc $(i, i+1)$ (see for example pages 61 et seq. for λ_i^1, λ_i^2 and λ_i^3).

Sometimes, we need to express that a binary variable δ is equal to one if and only if the integer variable $x \in \mathbb{N}_0$ is greater than zero and δ is equal to zero, else.

$$x > 0 \Leftrightarrow \delta = 1$$

This is ensured by

$$M\ \delta \geq x \tag{f1'}$$
$$\delta \leq x \tag{f2'}$$

where M is an upper bound on x. This is required, for example, for determining the binary decision variable $\alpha_{(i,i+1)}^{rest}$ which depends on the integer variable $\Delta_{(i,i+1)}^{rest}$ and vice versa on page 44.

The modeling techniques introduced will be used to develop the MILP models.

6.2 Parameters of the model

$r \in \mathbb{N}$	Total number of vertices. The vertices are numbered from 0 to $r-1$ according to the sequence of customer locations to be visited. The first vertex (0) represents the start position, the last vertex $(r-1)$ represents the last location (see Figure 3)
$\bar{\Delta}_{(i,i+1)}^{drive} \in \mathbb{N}_0$	Driving time in minutes needed to travel from i to $i+1$, $i = 0, \ldots, r-2$
$\bar{\Delta}_i^{service} \in \mathbb{N}_0$	Time needed for loading and/or unloading at vertex i, $i = 0, \ldots, r-1$, in minutes, $\bar{\Delta}_0^{service} = 0$
$nTW_i \in \mathbb{N}$	Number of time windows at customer location i, $i = 1, \ldots, r-1$
$\overline{TW}_{iz}^{begin} \in \mathbb{N}_0$	Lower limit of the time window z at vertex i, $i = 1, \ldots, r-1$,

	$z = 1, \dots, nTW$ in minutes counted from start time 0
$\overline{TW}^{end}_{iz} \in \mathbb{N}_0$	Upper limit of the time window z at vertex i, $i = 1, \dots, r-1$, $z = 1, \dots, nTW_i$ in minutes counted from start time 0
$\overline{udt} \in \mathbb{N}_0$	Driving time since the last daily rest period or break at the beginning of the planning horizon in minutes
$\overline{ddt} \in \mathbb{N}_0$	Cumulated daily driving time since the end of the last daily rest period at the beginning of the planning horizon in minutes
$\overline{ptr} \in \mathbb{N}_0$	Passed time since the end of the last daily rest period at the beginning of the planning horizon in minutes
$\overline{ptwr} \in \mathbb{N}_0$	Passed time since the end of the last weekly rest period at the beginning of the planning horizon in minutes
$\overline{urt} \in \mathbb{N}_0$	If a daily rest period takes place at start time, this parameter expresses its duration since the start of the rest period in minutes
$\overline{ubt} \in \mathbb{N}_0$	If a break takes place at start time, this parameter expresses its duration since the start of the break in minutes
$\overline{dte} \in \{0,1\}$	Is equal to 1 if a driving time extension is currently used when the planning horizon begins, 0 otherwise
$\overline{hpb} \in \{0,1\}$	Is equal to 1 if the first part of a break with a duration of at least 15 minutes has already been taken before the beginning of the planning horizon, 0 otherwise
$\overline{hpr} \in \{0,1\}$	Is equal to 1 if the first part of a daily rest period with a duration of at least 3 hours has already been taken before the beginning of the planning horizon, 0 otherwise
$\overline{noRed} \in \{0,1,2,3\}$	The number of reduced daily rest periods that have already been taken in the current week
$\overline{noExt} \in \{0,1,2\}$	The number of extended daily driving times that have already been taken in the current week

6.3 Variables

Variables needed to define the objective function

$start_i \in \mathbb{R}_0^+$	Start of service time in vertex $i, i = 1, ..., r-1$, Start of driving (after potential break or rest) for $i = 0$
$\Delta_i^{late} \in \mathbb{R}_0^+$	Lateness in vertex i, $i = 1, \ldots, r-1$

Variables that indicate which time window is chosen at customer location i

$$tw_{iz} = \begin{cases} 1 & \text{if time window z is chosen at destination } i \\ 0 & \text{otherwise} \end{cases}$$

$i = 1, \ldots, r-1, z = 1, \ldots, nbTW_i$

The following set comprises the continuous status variables for each vertex i.

E_i^{dt}	Driving time left until the next break or daily rest period when entering vertex i, $i = 0, \ldots, r-1$ in minutes $0 \leq E_i^{dt} \leq 270$
E_i^{ddt}	Driving time left until the next daily rest period when entering vertex i $i = 0, \ldots, r-1$ in minutes $0 \leq E_i^{ddt} \leq 540$
E_i^{t}	Time left until the next daily rest period when entering vertex i $i = 0, \ldots, r-1$ in minutes $0 \leq E_i^{t} \leq 900$
L_i^{dt}	Driving time left until the next break or daily rest period when leaving vertex i, $i = 0, \ldots, r-1$ in minutes $0 \leq L_i^{dt} \leq 270$
L_i^{ddt}	Driving time left until next daily rest when leaving vertex i $i = 0, \ldots, r-1$ in minutes $0 \leq L_i^{ddt} \leq 540$
L_i^{t}	Time left until the next daily rest period when leaving vertex i $i = 0, \ldots, r-1$ in minutes $0 \leq L_i^{t} \leq 900$

The following variables indicate for each arc $(i, i+1)$ if a daily rest is made, the number of daily rests and their cumulated duration.

$$\alpha_{(i,i+1)}^{rest} = \begin{cases} 1 & \text{if at least one daily rest is taken on arc } (i, i+1) \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-2$

$A^{rest}_{(i,i+1)} \in \mathbb{N}_0$ The number of daily rest periods taken on arc $(i, i+1)$, $i = 0, \ldots, r-2$

$\Delta^{rest}_{(i,i+1)} \in \mathbb{R}^+_0$ The cumulated duration of all daily rest periods on arc $(i, i+1)$, $i = 0, \ldots, r-2$

Regarding daily rests at vertices, the following variables indicate if a daily rest is made and its duration.

$$\alpha^{rest}_i = \begin{cases} 1 & \text{if a daily rest is made in vertex } i \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

$\Delta^{rest}_i \in \mathbb{R}^+_0$ The duration of a daily rest in vertex i, $i = 0, \ldots, r-1$

The next set of variables are needed to determine if breaks are taken on arc $(i, i+1)$ and their number.

$$\alpha^{break}_{(i,i+1)} = \begin{cases} 1 & \text{if at least one break is taken on arc } (i, i+1) \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-2$

$A^{break}_{(i,i+1)} \in \mathbb{N}_0$ The number of breaks taken on arc $(i, i+1)$, $i = 0, \ldots, r-2$

The following variables indicate if breaks are taken in vertices.

$$\alpha^{break}_i = \begin{cases} 1 & \text{if a break is taken in vertex } i \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

Each variable Δ^{wait}_i gives the waiting time in vertex i:

$\Delta^{wait}_i \in \mathbb{R}^+_0$ Waiting time in vertex i, $i = 0, \ldots, r-1$

The next variables specify if an early daily rest is made on an arc, meaning that the daily driving time is not completely used up.

$$\mu^{earlydr1}_{(i,i+1)} = \begin{cases} 1 & \text{if a break is replaced by a daily rest on arc } (i, i+1) \\ & \text{and this rest is the first rest on this arc} \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-2$

$$\mu_{(i,i+1)}^{earlydr2} = \begin{cases} 1 & \text{if a break is replaced by a daily rest on arc } (i, i+1) \\ & \text{and this rest is } \textit{not} \text{ the first rest on this arc} \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-2$

When arriving in vertex i, in case a daily rest was taken on arc $(i-1, i)$, the following variable indicates if a break was taken since the last daily rest.

$$e_i^{bt} = \begin{cases} 1 & \text{if the last rest activity on the preceding arc } (i-1, i) \text{ was a break} \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

The next variables indicate if a break is still necessary to completely use up the daily driving time left when leaving vertex i.

$$l_i^{bn} = \begin{cases} 1 & \text{if a break would be necessary to completely exploit} \\ & \text{the daily driving time left when leaving vertex } i \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

The following variables are needed to model the optional rules.

$$\alpha_i^{pbreak} = \begin{cases} 1 & \text{if the first part of a break is taken in vertex } i \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

$$\mu_{(i,i+1)}^{upbreak} = \begin{cases} 1 & \text{if the second part of a break is taken on arc } (i, i+1) \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-2$

$$\mu_i^{upbreak} = \begin{cases} 1 & \text{if the second part of a break is taken in vertex } i \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

$$l_i^{pbreak} = \begin{cases} 1 & \text{if when leaving vertex } i \text{ a partial break of 15 minutes was taken} \\ & \text{since the last rest period} \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

$$\alpha_i^{prest} = \begin{cases} 1 & \text{if the first part of a daily rest is taken in vertex } i \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

$$l_i^{prest} = \begin{cases} 1 & \text{if when leaving vertex } i \text{ a partial rest period of 3 h was taken} \\ & \text{since the last rest period} \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

$$\mu_i^{prest} = \begin{cases} 1 & \text{if the last break on arc } (i-1, i) \text{ is substituted by a} \\ & \text{first partial rest} \\ 0 & \text{otherwise} \end{cases}$$

$i = 1, \ldots, r-1$

$$\mu_i^{dredrest} = \begin{cases} 1 & \text{if in vertex } i \text{ the decision is made that the next} \\ & \text{daily rest after leaving vertex } i \text{ will be a reduced one} \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

$\mu_{(i,i+1)}^{redrest} \in \{0, 1, 2, 3\}$ The number of reduced daily rests made on arc $(i, i+1)$, $i = 0, \ldots, r-2$

$$\mu_i^{redrest} = \begin{cases} 1 & \text{if a reduced daily rest is taken in vertex } i \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

$$l_i^{dredrest} = \begin{cases} 1 & \text{if the next daily rest is a reduced one and is taken} \\ & \text{after leaving vertex } i \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

$$\mu_{(i,i+1)}^{extd1} = \begin{cases} 1 & \text{if a driving time extension is used on arc } (i, i+1) \text{ before the} \\ & \text{first daily rest} \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-2$

$\mu^{extd2}_{(i,i+1)} \in \{0,1,2\}$ The number of driving time extensions used on arc $(i, i+1)$ between the first and the last daily rest, $i = 0, \ldots, r-2$

$$\mu^{extd3}_{(i,i+1)} = \begin{cases} 1 & \text{if a driving time extension is used on arc } (i,i+1) \text{ after the} \\ & \text{last daily rest} \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-2$

$$\mu^{extd}_{i} = \begin{cases} 1 & \text{if a driving time extension is decided in vertex } i \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

$$l^{extd}_{i} = \begin{cases} 1 & \text{if a decision concerning a driving time extension was made} \\ & \text{before leaving vertex } i \\ 0 & \text{otherwise} \end{cases}$$

$i = 0, \ldots, r-1$

Auxiliary variables:

$$\begin{aligned} \lambda^1_i,\ \lambda^2_i,\ \lambda^3_i,\ \lambda^4_i,\ \lambda^6_i,\ &\in\ \{0,1\}, \qquad i = 0,\ldots,r-1 \\ \lambda^5_i\ &\in\ \{0,1\}, \qquad i = 0,\ldots,r-2 \end{aligned}$$

6.4 Optional rules

There are four optional rules that provide more flexibility to the driver schedule considering a planning period of one week (see page 7). According to the EC regulation, breaks and rest periods may be split into two parts (optional rules 1 and 2). The daily driving time may be extended from 9 to 10 hours two times a week (optional rule 4). The maximum time between the end of a daily rest period and the beginning of the next daily rest period may be extended from 13 to 15 hours (optional rule 3). Each possibility has to be considered when modeling the driver activities and determining the resulting driver status when entering and leaving a vertex.

The impact of each optional rule and the variables and constraints will be described next in more detail. This section will end with a description and modeling of dependencies of the different optional rules.

6.4.1 Splitting breaks

A break may be split into two parts, the first having a duration of at least 15 minutes (first partial break), and the second (second partial break) having a duration of at least 30 minutes. After the second part of the break, a new driving time interval of at most 4.5 hours starts.

Without loss of generality we assume that if a break is split in an optimal solution, the first partial break is taken to compensate waiting time at a customer location, i.e. we only allow first partial breaks in vertices and not on arcs. If we allowed a first partial break to be taken on an arc, either it could have been postponed to the next vertex or the second part of the break would also have to be taken on the same arc without impact on the driver status. Nevertheless, in practice, a complete break scheduled between two consecutive customer locations may be split by the driver without influencing the schedule.

We introduce the variable α_i^{pbreak} to indicate if a first partial break is taken upon arrival at a customer location i.

Taking into account the driver status and the driving time needed to visit the next customer, a break may be necessary on the way from customer i to customer $i+1$. If a first partial break has already taken place, only a second partial break needs to be scheduled. The variable $\mu_{(i,i+1)}^{upbreak}$ indicates that a second partial break is scheduled on arc $(i, i+1)$, i.e. the first break scheduled on this arc has a duration of 30 minutes instead of 45 minutes.

In case no break and no daily rest period are needed to traverse the arc $(i, i+1)$, a second partial break may also be taken on the following arc. A second partial break may also be scheduled in vertex $i+1$ to again compensate for waiting time and to allow for a new driving time interval to start. Variable $\mu_i^{upbreak}$ indicates whether a second partial break is made in a vertex. If no second partial break is scheduled, neither on arc $(i, i+1)$ nor in vertex $i+1$, the break may be completed on a subsequent arc or in a subsequent vertex and so on. To recall that a first partial break still may be used, we introduce the variable l_i^{pbreak} for each customer vertex. If l_i^{pbreak} equals one, the next break will only have to take 30 minutes (second part) instead of 45 minutes.

In detail, the conditions are as follows.

Only the second partial break has to be taken on arc $(i, i+1)$ instead of a full 45-minute break ($\mu_{(i,i+1)}^{upbreak} = 1$) if and only if a first partial break was taken before and no second partial break was made, yet (i.e. $l_i^{pbreak} = 1$). This means that we wish to impose that

$$\mu_{(i,i+1)}^{upbreak} = 1 \Leftrightarrow l_i^{pbreak} = 1 \wedge \alpha_{(i,i+1)}^{break} = 1$$

This is ensured by the following conditions:

$$\begin{aligned}
&(\mu_{(i,i+1)}^{upbreak} = 1 \Rightarrow l_i^{pbreak} = 1) && \wedge \\
&(\mu_{(i,i+1)}^{upbreak} = 1 \Rightarrow \alpha_{(i,i+1)}^{break} = 1) && \wedge \\
&(\mu_{(i,i+1)}^{upbreak} = 0 \Rightarrow \alpha_{(i,i+1)}^{break} = 0 \vee l_i^{pbreak} = 0) &&
\end{aligned}$$

Note that the last implication is the equivalent contraposition of ($l_i^{pbreak} = 1 \ \wedge \ \alpha_{(i,i+1)}^{break} = 1 \Rightarrow \mu_{(i,i+1)}^{upbreak} = 1$). For our MILP model, we obtain the following constraints:

$$\mu_{(i,i+1)}^{upbreak} \leq l_i^{pbreak} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.1}$$
$$\mu_{(i,i+1)}^{upbreak} \leq \alpha_{(i,i+1)}^{break} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.2}$$
$$\mu_{(i,i+1)}^{upbreak} \geq l_i^{pbreak} + \alpha_{(i,i+1)}^{break} - 1 \qquad \forall\ i = 0, \ldots, r-2 \tag{6.3}$$

A second partial break can also be taken upon the arrival at a customer location if the partial break status at the preceding vertex equals one and no break was taken on the preceding arc. Therefore, we want to state that

$$\mu_{i+1}^{upbreak} = 1 \Leftrightarrow l_i^{pbreak} = 1 \wedge \mu_{(i,i+1)}^{upbreak} = 0 \wedge \alpha_{i+1}^{break} = 1$$

This is achieved by adding the following constraints, again making use of a contraposition for the formulation of the last set of constraints:

$$\mu_{i+1}^{upbreak} \leq l_i^{pbreak} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.4}$$
$$\mu_{i+1}^{upbreak} \leq 1 - \mu_{(i,i+1)}^{upbreak} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.5}$$
$$\mu_i^{upbreak} \leq \alpha_i^{break} \qquad \forall\ i = 0, \ldots, r-1 \tag{6.6}$$
$$\mu_{i+1}^{upbreak} \geq l_i^{pbreak} - \mu_{(i,i+1)}^{upbreak} + \alpha_{i+1}^{break} - 1 \qquad \forall\ i = 0, \ldots, r-2 \tag{6.7}$$

Note that inequalities (6.4) to (6.7) also hold for vertex 0. The status variable l_i^{pbreak} indicating whether a first partial break still counts for the subsequent arc can now easily be determined by the following set of constraints.

$$l_{i+1}^{pbreak} = l_i^{pbreak} - \mu_{(i,i+1)}^{upbreak} - \mu_{i+1}^{upbreak} + \alpha_{i+1}^{pbreak} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.8}$$

For determining the status variable for partial breaks for the first vertex, we need to know if a partial break already took place. The input parameter $\overline{hpb}$ serves as an indicator if a partial break already took place before the starting time of the schedule. It is needed when the schedule does not start at the beginning of a week (i.e. after a weekly rest period), and it is equal to one if a partial break was already taken and zero else. If $\overline{hpb}$ is equal to one, a second partial break may be scheduled in the starting vertex. In that case, $\mu_i^{upbreak}$ is equal to one. If $\overline{hpb}$ is equal to zero, a first partial break may be taken in vertex one. Observe that α_0^{pbreak} and $\mu_0^{upbreak}$ cannot be both equal to one due to (6.6) and the vertex activity constraints (6.99) (see page 48).

The idea now is to define l_i^{pbreak} in such a way that a first partial break (except for a partial break taken before the planning horizon starts) always has to be exploited in the course of time. As it is possible that a first partial break has been taken before the beginning of the planning horizon, maybe it will not be beneficial to force its use. This would for example be the case if a daily rest would be necessary or advantageous to be taken before or instead of the next break. Therefore, only an upper and a lower bound for l_0^{pbreak} are given which induce the following logical conditions:

$$((\overline{hpb} = 0 \wedge \alpha_0^{pbreak} = 0) \vee \mu_0^{upbreak} = 1) \Rightarrow l_0^{pbreak} = 0$$
$$\alpha_0^{pbreak} = 1 \Rightarrow l_0^{pbreak} = 1$$

The corresponding upper and lower bounds are given by:

$$l_0^{pbreak} \leq \overline{hpb} + \alpha_0^{pbreak} \tag{6.9}$$
$$l_0^{pbreak} \leq 1 - \mu_0^{upbreak} \tag{6.10}$$
$$l_0^{pbreak} \geq \alpha_0^{pbreak} \tag{6.11}$$

We force a first partial break only to be scheduled if the corresponding second partial break is also scheduled to only keep track of first partial breaks that are necessary. l_i^{pbreak} has been introduced to allow us to not only consider just the arc or vertex directly after a first partial break has been taken. If $l_i^{pbreak} = 1$ and no second partial break is necessary on arc $(i, i+1)$ or at vertex $i+1$, l_{i+1}^{pbreak}again is equal to 1 and a second partial break is possible on arc $(i+1, i+2)$ or in vertex $i+2$ if a break is needed, and so on.

We can distinguish between four cases for which a first partial break will be of no use, as the second part will never be scheduled.

Case 1: It is not necessary to schedule a break to fully exploit the daily driving time left and no use is made of a driving time extension. l_i^{bn} indicates if a break is necessary to completely use up the daily driving time left (without extension) when leaving vertex i. $\mu_{(i,i+1)}^{extd1}$ indicates if a driving time extension is planned to take place on arc $(i, i+1)$ before the first daily rest. Observe that in that case an additional break would be necessary. We obtain the following logical conditions:

$$l_i^{bn} = 0 \wedge \mu_{(i,i+1)}^{extd1} = 0 \Rightarrow l_i^{pbreak} = 0$$
$$l_{r-1}^{pbreak} = 0$$

We add the following constraints to our model.

$$l_i^{pbreak} \leq l_i^{bn} + \mu_{(i,i+1)}^{extd1} \qquad \forall i = 0, \dots, r-2 \tag{6.12}$$
$$l_{r-1}^{pbreak} = 0 \tag{6.13}$$

Case 2: A break is replaced by a daily rest on arc $(i, i+1)$ and this rest is the first one on this arc (i.e. $\mu_{(i,i+1)}^{earlydr1} = 1$), which means, there will be no break before the next daily rest is taken. Hence, the second partial break again would never be scheduled.

The logical expression

$$\mu_{(i,i+1)}^{earlydr1} = 1 \Rightarrow l_i^{pbreak} = 0$$

is transformed into

$$l_i^{pbreak} \leq 1 - \mu_{(i,i+1)}^{earlydr1} \qquad \forall\, i = 0, \ldots, r-2 \tag{6.14}$$

Cases 3 and 4 refer to those situations in which a daily rest period ($\alpha_i^{rest} = 1$) or a partial daily rest period ($\alpha_i^{prest} = 1$) in vertex i avoid scheduling a second partial break. Observe that even if $\alpha_i^{prest} = 0$, in case that the last break on arc $(i-1, i)$ is substituted by a partial rest (indicated by variable $\mu_i^{prest} = 1$)[1], the status variable l_i^{pbreak} may still obtain the value 1, as a first partial break may be scheduled for a driving time extension. We transform the logical conditions

$$\begin{aligned} &\alpha_i^{rest} = 1 \Rightarrow l_i^{pbreak} = 0 \quad \wedge \\ &(\alpha_i^{prest} = 1 \wedge \mu_i^{prest} = 0) \Rightarrow l_i^{pbreak} = 0 \end{aligned}$$

into the following constraints:

$$l_i^{pbreak} \leq 1 - \alpha_i^{rest} \qquad \forall\, i = 0, \ldots, r-2 \tag{6.15}$$

$$l_i^{pbreak} \leq 1 - \alpha_i^{prest} + \mu_i^{prest} \qquad \forall\, i = 0, \ldots, r-2 \tag{6.16}$$

6.4.2 Splitting daily rest periods

A daily rest period may be split into two parts, one having a duration of at least 3 hours (first partial rest), and the other (second partial rest) with a duration of at least 9 hours. After the second partial rest, the daily driving time left is reset to 9 hours.

Splitting a daily rest period is advantageous if the first partial rest period compensates waiting time at a customer location and is similar to splitting breaks (see Figure 9, (2)).

In addition, instead of scheduling a first partial rest period in vertex i, it is also possible to substitute the last break on the preceding arc by a first partial rest ($\mu_i^{prest} = 1$). As a partial rest also resets the 4.5 hour driving time interval, a 45 minute break can then be left out (see Figure 9, (3)). We will consider this in more detail at the end of this section (see constraints (6.24) to (6.30) for μ_i^{prest}).

We introduce the variable α_i^{prest} to indicate whether a partial break is made upon arrival at customer location i. If $\alpha_i^{prest} = 1$, then it may be necessary to schedule the second partial rest on the way from customer i to customer $i+1$. The driver status as well as the driving time needed from i to $i+1$ determine whether a daily rest period is required on the arc $(i, i+1)$. In case no daily rest period is needed, the first partial rest still "counts" for the subsequent arc. To keep track if a first partial rest still may be used, we introduce the variable l_i^{prest}. If $l_i^{pbreak} = 1$ then the next daily rest period will only have a duration of at least 9 hours instead of 11 hours.

In contrast to the case of splitting breaks, it is not necessary to define a variable to indicate if a second partial rest period is scheduled. As sometimes it may be beneficial to take a

[1] The partial rest period is taken instead of the last break on the arc $(i-1, i)$ and not in vertex i if $\mu_i^{prest} = 1$.

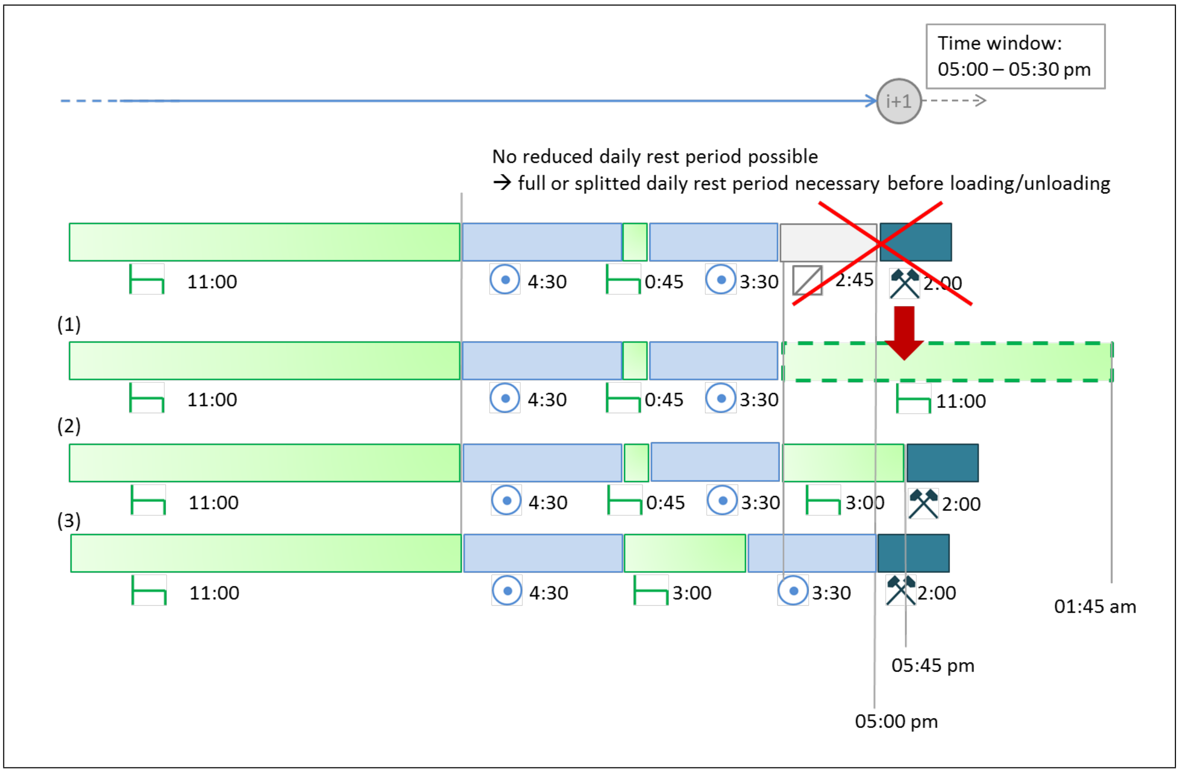

Figure 9: Splitting daily rest periods

daily rest that lasts longer than the minimum duration of 9 or 11 hours, respectively, variables are introduced to identify the actual duration instead (see page 43, constraints (6.79) to (6.84)).

Also different from partial breaks, it may be advantageous to take a first partial rest for which the second part will be scheduled after the planning horizon as the decision about taking a first partial rest will influence the time left until the next daily rest period is necessary. If, for example, a first partial rest is taken at the last customer location, the remaining duration of the second partial rest period is at least 9 hours instead of 11 hours. In that case, more time will be left to fulfill the service at the customer location[1].

Let us now consider the status variable l_i^{prest}, which indicates whether only the second part of a daily rest period is still needed to reset the time left until the next daily rest period. If a daily rest is taken in vertex i, the status variable l_i^{prest} is set to zero as a potential first

[1] Assume that in Figure 9 the location $i+1$ is the last customer location and no reduced daily rest period is possible anymore (i.e. in the current week 3 reduced daily rest periods have already taken place). The driver would reach the time window, but adding 2 hours for serving the customer would lead to a working time of 13:30 h since the last daily rest period. This violates rule 6 as it is not possible to append a daily rest period of 11 h and finish it within the 24 h time interval. A daily rest period would be necessary before loading and/or unloading which would have to be postponed to the following day.

partial rest would have been used to reduce the duration of this daily rest period. Hence,

$$\alpha_i^{rest} = 1 \Rightarrow l_i^{prest} = 0$$

which is expressed by

$$l_i^{prest} \leq 1 - \alpha_i^{rest} \qquad \forall\ i = 0, \dots, r-1 \tag{6.17}$$

Clearly, l_i^{prest} should always be equal to 1 if a first partial rest is taken in vertex i. That means

$$\alpha_i^{prest} = 1 \Rightarrow l_i^{prest} = 1$$

This is induced by

$$l_i^{prest} \geq \alpha_i^{prest} \qquad \forall\ i = 0, \dots, r-1 \tag{6.18}$$

If a daily rest is taken on arc $(i, i+1)$, the status variable l_{i+1}^{pbreak} is only influenced by a potential partial rest in vertex $i+1$. We would like to state

$$\alpha_{(i,i+1)}^{rest} = 1 \Rightarrow l_{i+1}^{prest} = \alpha_{i+1}^{prest}$$

This is imposed by inequalities (6.18) and the following set of constraints:

$$l_{i+1}^{prest} \leq \alpha_{i+1}^{prest} + 1 - \alpha_{(i,i+1))}^{rest} \qquad \forall\ i = 0, \dots, r-2 \tag{6.19}$$

An upper bound on the value of the status variable l_{i+1}^{prest} is $l_i^{prest} + \alpha_{i+1}^{prest}$. If both variables l_i^{prest} and α_{i+1}^{prest} are zero then l_{i+1}^{prest} must also be equal to zero:

$$l_i^{prest} = 0 \wedge \alpha_{i+1}^{prest} = 0 \Rightarrow l_{i+1}^{pbreak} = 0$$

This is induced by

$$l_{i+1}^{prest} \leq l_i^{prest} + \alpha_{i+1}^{prest} \qquad \forall\ i = 0, \dots, r-2 \tag{6.20}$$

If neither a daily rest was taken on arc $(i, i+1)$ nor in vertex $i+1$, then l_{i+1}^{pbreak} depends on both variables, l_i^{prest} and α_{i+1}^{prest}. In this case, we have

$$(\alpha_{(i,i+1)}^{rest} = 0 \wedge \alpha_{i+1}^{rest} = 0) \Rightarrow (l_{i+1}^{prest} = 1 \Leftrightarrow (l_i^{prest} = 1 \vee \alpha_{i+1}^{prest} = 1))$$

This is ensured by constraints (6.20) and

$$l_{i+1}^{prest} \geq l_i^{prest} + \alpha_{i+1}^{prest} - \alpha_{i+1}^{rest} - \alpha_{(i,i+1)}^{rest} \qquad \forall\ i = 0, \dots, r-2 \tag{6.21}$$

Note that inequalities (6.21) enforce $l_i^{prest} + \alpha_{i+1}^{prest} \leq 1$ in the case that no daily rest on the arc $(i, i+1)$ or in the vertex i is made. This means that a first partial rest period has to be exploited by taking a second partial daily rest period before a new first partial daily rest period can be taken.

In the first vertex, l_0^{prest} only depends on α_0^{prest} and if a partial rest was made before the start of the planning horizon ($\overline{hpr} = 1$) if no rest is made in 0. The case that a rest is made is already considered by constraints (6.17).

$$\alpha_0^{rest} = 0 \Rightarrow l_0^{prest} = \alpha_0^{prest} + \overline{hpr}$$

As $\alpha_0^{prest} + \overline{hpr}$ is an upper bound on l_0^{prest}, we obtain the following constraints:

$$l_0^{prest} \geq \alpha_0^{prest} + \overline{hpr} - \alpha_0^{rest} \tag{6.22}$$
$$l_0^{prest} \leq \alpha_0^{prest} + \overline{hpr} \tag{6.23}$$

As mentioned at the beginning of this section instead of taking a break as the last resting activity on arc $(i, i+1)$, it may also be possible to substitute this break by a first partial rest period. The first partial rest also resets the driving time left until the next break or rest period, so it may save time to just schedule a first partial rest period on the arc $(i, i+1)$ as the last resting activity instead of a 45-minute break and additionally a first partial daily rest in vertex $i+1$ (see Figure 9 (3)). We introduce the variable μ_i^{prest} to indicate that a substitution takes place and the status variables when entering vertex i and the arrival time at customer i have to be modified accordingly. If such a substitution is planned, a first partial rest period has to be scheduled:

$$\mu_i^{prest} = 1 \Rightarrow \alpha_i^{prest} = 1$$

This logical condition is represented by

$$\alpha_i^{prest} \geq \mu_i^{prest} \qquad \forall\ i = 1, \dots, r-1 \tag{6.24}$$

For vertex 0, since there is no preceding arc where we can substitute a break, we have

$$\mu_0^{prest} = 0 \tag{6.25}$$

For other vertices i, there has to be a break on arc $(i-1, i)$ that may be substituted and this break has to be the last resting activity on this arc. When entering vertex i, variable e_i^{bt} indicates if the last rest activity taken on the arc $(i-1, i)$ was a break. We obtain the following upper bounds on μ_i^{prest}:

$$\mu_i^{prest} \leq \alpha_{(i-1,i)}^{break} \qquad \forall\ i = 1, \dots, r-1 \tag{6.26}$$
$$\mu_i^{prest} \leq e_i^{bt} \qquad \forall\ i = 1, \dots, r-1 \tag{6.27}$$

If the daily driving time should be extended and the corresponding resting activity is planned to be a first partial rest in a vertex, a preceding break on arc $(i, i+1)$ will not be substituted, as both rest activities (45-minute break and first partial rest) are needed to be able to extend the daily driving time to 10 hours. Therefore, we obtain the upper bounds:

$$\mu_i^{prest} \leq 1 - \mu_i^{extd} \qquad \forall\ i = 1, \dots, r-1 \tag{6.28}$$

Whenever possible, the substitution should take place, as it will never worsen the objective function value and it may save up to 45 minutes of time. Whenever possible means, if a break is made on arc $(i-1,i)$ ($\alpha^{break}_{(i-1,i)} = 1$), the last resting activity on this arc was a break, and a first partial rest is associated with vertex i ($\alpha^{prest}_i = 1$), then the break will be substituted if no break for a driving time extension is needed in vertex i. This is expressed by

$$(\alpha^{break}_{(i-1,i)} = 1 \wedge e^{bt}_i = 1 \wedge \mu^{extd}_i = 0) \Rightarrow \mu^{prest}_i = \alpha^{prest}_i$$

and guaranteed by inequalities (6.24) and the following constraints

$$\mu^{prest}_i \geq \alpha^{prest}_i - \mu^{extd}_i - (1-\alpha^{break}_{(i-1,i)}) - (1-e^{bt}_i) \qquad \forall\, i = 1,\dots,r-1 \tag{6.29}$$

If there is only one break on arc $(i-1,i)$, a potential first partial break still counting when leaving vertex $i-1$ may be consumed by the substitution and would therefore not be necessary to be scheduled. To avoid this, we add the following upper bound on l^{pbreak}_i with $A^{break}_{(i-1,i)}$ being the number of breaks scheduled on arc $(i-1,i)$:

$$l^{pbreak}_{i-1} \leq A^{break}_{(i-1,i)} - \mu^{prest}_i + (1-\alpha^{break}_{(i-1,i)}) \qquad \forall\, i = 1,\dots,r-1 \tag{6.30}$$

6.4.3 Reducing daily rest periods

A daily rest period may be reduced from 11 to 9 hours at most three times a week. If it is reduced, the time between the end of a daily rest period and the beginning of the subsequent reduced daily rest period is automatically extended from 13 to 15 hours because of standard rule 6 (page 6), which states that there has to be a daily rest period in each time interval with a duration of 24 hours.

The difficulty here lies in the fact that after a daily rest period, depending on whether the next daily rest period is a reduced one or not, the time interval between the two rest periods has a duration of 13 or 15 hours, respectively.

To overcome this, for the first daily rest on an arc, we introduce the variable $\mu^{dredrest}_i$ associated with the decision about the next rest after leaving vertex i being a reduced one. Additionally, the status variable $l^{dredrest}_i$ monitors if a decision about a reduced daily rest period was made before leaving vertex i to keep track of a decision made about a reduced daily rest period at a vertex prior to i if no rest period was scheduled since then.

If a second or third daily rest is made on an arc between two customers, only the driving time has to be considered. As there are always at least 13 hours between two daily rests but only at most 10 hours of driving allowed, no special care has to be taken about a reduced daily rest. However, its duration needs to be modified.

The variable $\mu^{redrest}_{(i,i+1)}$ gives the number of reduced rest periods scheduled for the arc $(i,i+1)$, while the variable $\mu^{redrest}_i$ indicates if a reduced rest period is scheduled in vertex i. These variables are used to modify the duration of daily rest periods such that instead of 11 hours at least 9 hours are needed.[1] This means that the reduction has to be scheduled together with the daily rest period itself, namely

$$(\mu_i^{redrest} = 1 \Rightarrow \alpha_i^{rest} = 1) \quad \wedge \quad (\mu_{(i,i+1)}^{redrest} = k \Rightarrow A_{(i,i+1)}^{rest} \geq k), \quad k \in \{0, 1, 2, 3\}$$

The above conditions are expressed by constraints (6.31) and (6.32).

$$\alpha_i^{rest} \geq \mu_i^{redrest} \qquad \forall\ i = 0, \ldots, r-1 \tag{6.31}$$

$$A_{(i,i+1)}^{rest} \geq \mu_{(i,i+1)}^{redrest} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.32}$$

If when leaving vertex i the decision is made that the next daily rest period will be a reduced one ($\mu_i^{dredrest} = 1$), the corresponding status variable ($l_i^{dredrest}$) is set to be equal to one. This means that

$$\mu_i^{dredrest} = 1 \Rightarrow l_i^{dredrest} = 1$$

and this is represented by

$$l_i^{dredrest} \geq \mu_i^{dredrest} \qquad \forall\ i = 0, \ldots, r-1 \tag{6.33}$$

If a daily rest is made either on arc $(i-1, i)$ or in vertex i, then 6.33 should hold as an equality, since in this case the status variable $l_i^{dredrest}$ only depends on the decision about a reduced rest period when leaving vertex i. The execution of former decisions about short rests lies in the past. This can be stated by

$$(\alpha_{(i-1,i)}^{rest} = 1 \Rightarrow l_i^{dredrest} = \mu_i^{dredrest}) \quad \wedge \quad (\alpha_i^{rest} = 1 \Rightarrow l_i^{dredrest} = \mu_i^{dredrest})$$

This is induced by constraints (6.33), (6.34) and (6.35)

$$l_i^{dredrest} \leq \mu_i^{dredrest} + (1 - \alpha_{(i-1,i)}^{rest}) \qquad \forall\ i = 1, \ldots, r-1 \tag{6.34}$$

$$l_i^{dredrest} \leq \mu_i^{dredrest} + (1 - \alpha_i^{rest}) \qquad \forall\ i = 0, \ldots, r-1 \tag{6.35}$$

In case neither a daily rest period was made on the arc $(i, i+1)$ nor in vertex $i+1$, the status variable $l_{i+1}^{dredrest}$ additionally depends on the status variable $l_i^{dredrest}$ of the preceding vertex i. Hence,

$$(\alpha_{(i,i+1)}^{rest} = 0 \wedge \alpha_{i+1}^{rest} = 0) \Rightarrow (l_{i+1}^{dredrest} = l_i^{dredrest} + \mu_{i+1}^{dredrest})$$

As $l_i^{dredrest} + \mu_{i+1}^{dredrest}$ is an upper bound on $l_{i+1}^{dredrest}$, we obtain the following constraints:

$$l_{i+1}^{dredrest} \leq l_i^{dredrest} + \mu_{i+1}^{dredrest} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.36}$$

$$l_{i+1}^{dredrest} \geq l_i^{dredrest} + \mu_{i+1}^{dredrest} - \alpha_{(i,i+1)}^{rest} - \alpha_{i+1}^{rest} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.37}$$

The status variable $l_0^{dredrest}$ of the starting vertex only depends on the decision about a reduced daily rest period:

$$l_0^{dredrest} = \mu_0^{dredrest} \tag{6.38}$$

[1] Restrictions on the duration of daily rest periods are described in more detail in Section 6.9 on page 43.

If several daily rest periods are scheduled on an arc $(i, i+1)$, the maximum time saving with a reduced daily rest can be made by planning the first daily rest period to be a reduced one. Therefore, in case at least one reduced daily rest is planned for $(i, i+1)$, we set $l_i^{dredrest} = 1$[1], except for the case a second partial rest period still has to be scheduled. Similarly, if a reduced daily rest is made in vertex $i+1$ and there were no daily rest periods on the previous arc $(i, i+1)$, the status variable $l_i^{dredrest}$ is set to be equal to 1 if $l_i^{prest} = 0$. The corresponding logical conditions are:

$$(\mu_{(i,i+1)}^{redrest} \geq 1 \wedge l_i^{prest} = 0) \Rightarrow l_i^{dredrest} = 1 \quad \wedge$$
$$(\mu_{i+1}^{redrest} = 1 \wedge \alpha_{(i,i+1)}^{rest} = 0) \Rightarrow l_i^{dredrest} = 1$$

The total number of reduced daily rest periods in one week is at most 3 and therefore, this is also an upper bound on the number of reduced daily rest periods on an arc. It follows that the above conditions are induced by

$$3\, l_i^{dredrest} \geq \mu_{(i,i+1)}^{redrest} - 3\, l_i^{prest} \qquad \forall\, i = 0, \ldots, r-2 \tag{6.39}$$
$$l_i^{dredrest} \geq \mu_{i+1}^{redrest} - \alpha_{(i,i+1)}^{rest} \qquad \forall\, i = 0, \ldots, r-2 \tag{6.40}$$

Additionally, we interlink the decision about a reduced daily rest period and the end of the last daily rest period to ensure that the status variables E_i^t, L_i^t, L_i^{ddt} and L_i^{dt} for vertex i reflect the exact driver status. The decision that the next daily rest should be a reduced one ($\mu_{i+1}^{dredrest} = 1$) can only be made if there was a daily rest period on the previous arc or vertex. This condition guarantees that decisions about reduced daily rest periods are made as early as possible such that status variables will reflect the actual driver status when entering or leaving a vertex. We add the conditions

$$(\alpha_{(i,i+1)}^{rest} = 0 \wedge \alpha_{i+1}^{rest} = 0) \Rightarrow \mu_{i+1}^{dredrest} = 0$$

and obtain the following set of inequalities:

$$\mu_{i+1}^{dredrest} \leq \alpha_{(i,i+1)}^{rest} + \alpha_{i+1}^{rest} \qquad \forall\, i = 0, \ldots, r-2. \tag{6.41}$$

Reduced rests can now be scheduled according to the status variables $l_i^{dredrest}$. That means, if at least one daily rest is scheduled on an arc $(i, i+1)$ and the status variable $l_i^{dredrest}$ is equal to 1, then at least one of the daily rests has to be a reduced one.

$$\alpha_{(i,i+1)}^{rest} = 1 \Rightarrow \mu_{(i,i+1)}^{redrest} \geq l_i^{dredrest}$$

This is ensured by

$$\mu_{(i,i+1)}^{redrest} \geq l_i^{dredrest} - (1 - \alpha_{(i,i+1)}^{rest}) \qquad \forall\, i = 0, \ldots, r-2 \tag{6.42}$$

If no rest is made on the arc $(i, i+1)$, a similar condition holds for the following vertex $i+1$:

$$\alpha_{(i,i+1)}^{rest} = 0 \Rightarrow (\alpha_{i+1}^{rest} = 1 \Rightarrow \mu_{i+1}^{redrest} \geq l_i^{dredrest})$$

[1] Note that reduced daily rest periods may be scheduled on arc $(i, i+1)$ if $l_i^{dredrest} = 0$. In that case, additional time prior to this reduced daily rest period is not needed, just the 2 hours time saving for the reduction of the duration is considered.

This is guaranteed by the following constraints.

$$\mu_{i+1}^{redrest} \geq l_i^{dredrest} - \alpha_{(i,i+1)}^{rest} - (1 - \alpha_{i+1}^{rest}) \qquad \forall\ i = 0, \ldots, r-2 \tag{6.43}$$

To ensure that the maximum number of reduced daily rest periods during one week is not exceeded, we add the constraint

$$\sum_{i=0}^{r-1} \mu_i^{redrest} + \sum_{i=0}^{r-2} \mu_{(i,i+1)}^{redrest} + \overline{noRed} \leq 3 \tag{6.44}$$

where $\overline{noRed}$ is the number of reduced daily rests already taken in the current week in the time before the start of the schedule.

We conclude this section by describing constraints that map dependencies between partial and reduced daily rest periods. First, we have to ensure that enough rest periods are scheduled, one for each decision about a reduced daily rest period and one for each second partial rest period. For the case that rest periods are taken on arc $(i, i+1)$, we obtain the condition

$$\alpha_{(i,i+1)}^{rest} = 1 \Rightarrow A_{(i,i+1)}^{rest} \geq \mu_{(i,i+1)}^{redrest} + l_i^{prest}$$

We transform this into

$$\mu_{(i,i+1)}^{redrest} + l_i^{prest} \leq A_{(i,i+1)}^{rest} + (1 - \alpha_{(i,i+1)}^{rest}) \qquad \forall\ i = 0, \ldots, r-2 \tag{6.45}$$

In addition, only one of the variables, $l_i^{dredrest}$ or l_i^{prest}, may take on the value 1. Both cannot be equal to 1 as otherwise, the next daily rest would be a second partial rest period and a reduced daily rest period as well, which does not make any sense.

$$(l_i^{dredrest} = 1 \Rightarrow l_i^{prest} = 0) \quad \wedge \quad (l_i^{prest} = 1 \Rightarrow l_i^{dredrest} = 0)$$

is imposed by

$$l_i^{dredrest} + l_i^{prest} \leq 1 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.46}$$

For the first vertex it has to be ensured that no reduced rest period is scheduled if a second partial rest is still outstanding. Hence,

$$\overline{hpr} = 1 \Rightarrow \mu_0^{redrest} = 0$$

This is guaranteed by the following constraint.

$$\mu_0^{redrest} \leq 1 - \overline{hpr} \tag{6.47}$$

6.4.4 Extending daily driving times

The daily driving time, i.e. the cumulated driving time between two consecutive daily rest periods, may be extended from 9 to 10 hours twice a week. If a daily driving time is extended, an additional break (or a first partial daily rest period) will be necessary, as there has to be a break after at most 4.5 hours of driving.

Driving time extensions have an impact on the values of the status variables E^t_{i+1}, L^t_{i+1}, E^{ddt}_{i+1}, L^{ddt}_{i+1}, E^{dt}_{i+1} and L^{dt}_{i+1} depending on when the additional break (or first partial daily rest period) is taken. We distinguish between the following four cases and introduce decision variables accordingly. The additional break can be taken:

case 1: in a vertex i ($\mu^{extd}_i = 1$),

case 2: on an arc $(i, i+1)$, before the first daily rest period is taken or if no daily rest period is taken on this arc ($\mu^{extd1}_{(i,i+1)} = 1$),

case 3: on an arc $(i, i+1)$, between two consecutive daily rest periods ($\mu^{extd2}_{(i,i+1)} = 1$) or

case 4: on an arc $(i, i+1)$, after the last daily rest ($\mu^{extd3}_{(i,i+1)} = 1$).

The main difference between scheduling a break or a first partial rest period for a driving time extension on an arc from scheduling it in a vertex is that they are always scheduled as late as possible (i.e. after 4.5 hours of driving, directly initiating the extension). In general, breaks can be scheduled in vertices to reduce or avoid waiting time, and therefore they may be taken before the limit for the driving time without break or rest period of 4.5 hours is reached.

Furthermore, on an arc a break or a first daily rest period for a driving time extension can be taken before the first daily rest period is completed (if there is a daily rest period on this arc) (case 1), between two consecutive daily rest periods (case 2) or after the last daily rest period (case 3).

In case 1, the time left until the next rest period when leaving vertex i, L^t_i, has to be taken into consideration, as this may limit the extended daily driving time to a value that is less than 10 hours. If a driving time extension occurs between two consecutive daily rest periods ($\mu^{extd2} = 1$) (case 2), then this will lead to the maximum daily driving time of 10 hours as the daily driving time is not limited by the maximum time interval between two consecutive daily rests. For the third case, we have to ensure that two breaks after the last daily rest period are scheduled such that the last of the two breaks coincides with the decision about the driving time extension.

Hence, if a special type of driving time extension can be scheduled on an arc, it depends on the number of daily rest periods on this arc. $\mu^{extd1}_{(i,i+1)}$ does not depend on whether one or more daily rest periods are made on arc $(i, i+1)$, whereas $\mu^{extd3}_{(i,i+1)} = 1$ requires at least one daily rest period. $\mu^{extd2}_{(i,i+1)}$ larger than 0 requires at least two daily rest periods on arc $(i, i+1)$, depending on the number of driving time extensions taken between the first and the last daily rest period.

We wish to state that

$$\mu^{extd2}_{(i,i+1)} = k \Rightarrow A^{rest}_{(i,i+1)} \geq k+1 \qquad \text{with } k \in \{0,1,2\} \qquad \text{and}$$
$$\mu^{extd3}_{(i,i+1)} = 1 \Rightarrow \alpha^{rest}_{(i,i+1)} = 1$$

This is achieved by the following constraints:

$$A^{rest}_{(i,i+1)} \geq \mu^{extd2}_{(i,i+1)} + \alpha^{rest}_{(i,i+1)} \qquad \forall\, i = 0,\dots,r-2 \tag{6.48}$$
$$\alpha^{rest}_{(i,i+1)} \geq \mu^{extd3}_{(i,i+1)} \qquad \forall\, i = 0,\dots,r-2 \tag{6.49}$$

In addition to the decision variables for extending driving, we again need status variables which in this case indicate if a driving time extension still holds when leaving vertex i (l^{extd}_i). This status variable ensures that a driving time extension cannot be made twice without a rest period in between.

We start with the corresponding constraint for vertex 0. The input data tells us if an extended daily driving time has already started: if the daily driving time $\overline{ddt}$ is greater than 540 and the number of extended driving times already taken that week is less than 2, the input parameter $\overline{dte}$ is set to be equal to 1, otherwise $\overline{dte} = 0$. We require that an already started driving time remains active if no daily rest is made in vertex 0:

$$\alpha^{rest}_0 = 0 \Rightarrow l^{extd}_0 = \overline{dte}$$

This is guaranteed by

$$l^{extd}_0 \geq \overline{dte} - \alpha^{rest}_0 \qquad \text{and} \tag{6.50}$$
$$l^{extd}_0 \leq \overline{dte} \tag{6.51}$$

No driving time extension needs to be scheduled in vertex 0 as such an extension can also be postponed to the arc $(0,1)$. Hence,

$$\mu^{extd}_0 = 0. \tag{6.52}$$

Before a daily rest is made on arc $(i, i+1)$, a driving time extension can only take place ($\mu^{extd1}_{(i,i+1)} = 1$) if no driving time extension is still active when leaving customer i. This is expressed by

$$l^{extd}_i = 1 \Rightarrow \mu^{extd1}_{(i,i+1)} = 0$$

We add the constraints

$$\mu^{extd1}_{(i,i+1)} \leq 1 - l^{extd}_i \qquad \forall\, i = 0,\dots,r-2 \tag{6.53}$$

to our model.

We wish to avoid that an early daily rest period is scheduled ($\mu^{earlydr1}_{(i,i+1)} = 1$) simultaneously with a driving time extension before the first daily rest period (if one is taken) on arc

$(i, i+1)$ is taken, since in reality this combination is impossible.[1] Therefore, the following condition must hold.

$$\mu^{earlydr1}_{(i,i+1)} = 1 \Rightarrow \mu^{extd1}_{(i,i+1)} = 0$$

This is imposed by constraints (6.54).

$$\mu^{extd1}_{(i,i+1)} \leq 1 - \mu^{earlydr1}_{(i,i+1)} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.54}$$

Usually, it does not improve the solution value to schedule a combination of extended driving times and early daily rests on one arc. However, there are two exceptions: the combination of an early daily rest as first daily rest on arc $(i, i+1)$ and an extended driving time of type two ($\mu^{extd2}_{(i,i+1)} \geq 1$) or of type three ($\mu^{extd3}_{(i,i+1)} = 1$). The other cases are avoided if the following conditions are imposed. The variable $\mu^{earlydr2}_{(i,i+1)} = 1$ indicates that an early daily rest period is taken and this daily rest period is not the first one on this arc.

$$\mu^{earlydr2}_{(i,i+1)} = 1 \Rightarrow (\mu^{extd1}_{(i,i+1)} = 0 \wedge \mu^{extd2}_{(i,i+1)} = 0 \wedge \mu^{extd3}_{(i,i+1)} = 0)$$

We add the following constraints:

$$\mu^{extd1}_{(i,i+1)} \leq 1 - \mu^{earlydr2}_{(i,i+1)} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.55}$$

$$\mu^{extd2}_{(i,i+1)} \leq 2 - 2\ \mu^{earlydr2}_{(i,i+1)} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.56}$$

$$\mu^{extd3}_{(i,i+1)} \leq 1 - \mu^{earlydr2}_{(i,i+1)} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.57}$$

In constraints (6.56), the upper bound 2 is the maximum number of driving time extensions allowed during one week.

A driving time extension $\mu^{extd1}_{(i,i+1)} = 1$ may not be possible if most of the available time between two daily rest periods was already spent for other activities different from driving. In that case, $\mu^{extd1}_{(i,i+1)}$ is set to be zero. The variable giving us information if a driving time extension is possible, considering the maximum time left until the start of the next daily rest period, is variable λ^1. If λ^1 is equal to 1, then no extension is possible, otherwise an extension is permitted:[2]

$$\lambda^1_i = 1 \Rightarrow \mu^{extd1}_{(i,i+1)} = 0$$

This is enforced by

$$\mu^{extd1}_{(i,i+1)} \leq 1 - \lambda^1_i \qquad \forall\ i = 0, \ldots, r-2 \tag{6.58}$$

For $\mu^{extd3}_{(i,i+1)}$, we wish to ensure that the driving time extension is coupled with a second break since the last daily rest period. This means that at least nine hours of daily driving time

[1] For more details on early daily rests see page 46.

[2] The auxiliary variable λ^1 is described in more detail on page 61.

have already been consumed when entering vertex $i+1$ since the MILP model schedules breaks and rest periods on arcs as late as possible. Therefore, at most one hour may be left for driving until the next rest period. Similarly, this is also true for $\mu^{extd1}_{(i,i+1)}$ if no daily rest period is taken on arc $(i,i+1)$. Hence, the following conditions must hold:

$$\begin{aligned} \mu^{extd3}_{(i,i+1)} = 1 \Rightarrow E^{ddt}_{i+1} \leq 60 \quad \wedge \\ (\mu^{extd1}_{(i,i+1)} = 1 \wedge \alpha^{rest}_{(i,i+1)} = 0) \Rightarrow E^{ddt}_{i+1} \leq 60 \end{aligned}$$

With 540 (i.e. 9 h) being an upper bound on E^{ddt}_{i+1}, we can express this condition using a "big-M approach" with $M = 480$ $(= 540 - 60)$.

$$E^{ddt}_{i+1} \leq 60 + 480\ (1 - \mu^{extd3}_{(i,i+1)}) \qquad \forall\ i = 0, \ldots, r-2 \tag{6.59}$$

$$E^{ddt}_{i+1} \leq 60 + 480\ (1 - \mu^{extd1}_{(i,i+1)}) + 480\ \alpha^{rest}_{(i,i+1)} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.60}$$

If there was no rest period since leaving the last vertex (i.e. $\alpha^{rest}_{(i,i+1)} = 0$) and a driving time extension was active when leaving that vertex ($l^{extd}_i = 1$), then this driving time extension is still active when entering $i+1$. As a result, another driving time extension is not allowed to start in vertex $i+1$. A new driving time extension may not start in $i+1$ if a driving time extension has already started on arc $(i,i+1)$ and no rest period was taken since then:

$$((l^{extd}_i = 1 \vee \mu^{extd1}_{(i,i+1)} = 1) \wedge \alpha^{rest}_{(i,i+1)} = 0) \Rightarrow \mu^{extd}_{i+1} = 0$$

This is imposed by

$$\mu^{extd}_{i+1} \leq 1 - l^{extd}_i + \alpha^{rest}_{(i,i+1)} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.61}$$

$$\mu^{extd}_{i+1} \leq 1 - \mu^{extd1}_{(i,i+1)} + \alpha^{rest}_{(i,i+1)} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.62}$$

For the case that an extended driving time has started on arc $(i,i+1)$ after the last daily rest period, i.e. $\mu^{extd3}_{(i,i+1)} = 1$, the above conditions are completed by adding the following one:

$$\mu^{extd3}_{(i,i+1)} = 1 \Rightarrow \mu^{extd}_{i+1} = 0$$

We obtain the additional constraints

$$\mu^{extd}_{i+1} \leq 1 - \mu^{extd3}_{(i,i+1)} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.63}$$

If the daily driving time left, E^{ddt}_i, is higher than 270 minutes (that is 4.5 h), a break in vertex i should not be considered to extend the daily driving time:

$$E^{ddt}_i > 270 \Rightarrow \mu^{extd}_i = 0$$

This is ensured by the following constraints:

$$270\ (1 - \mu^{extd}_i) \geq E^{ddt}_i - 270 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.64}$$

Now, the determination of the status variables l_{i+1}^{extd} when leaving vertex $i+1$ is described. If a daily rest period is taken in vertex i, then a new driving time interval starts. An extended driving time still active when entering vertex i is finished with the start of the daily rest period. Therefore, the status variable l_{i+1}^{extd} is set to be zero. If a daily rest period is taken on arc $(i,i+1)$, l_{i+1}^{extd} depends on $\mu_{(i,i+1)}^{extd3}$ and μ_{i+1}^{extd}. It depends on l_i^{extd} and $\mu_{(i,i+1)}^{extd1}$ and μ_{i+1}^{extd} if no daily rest is made on arc $(i,i+1)$. In particular, the following conditions have to hold:

$$\begin{aligned}
\alpha_i^{rest} = 1 &\Rightarrow l_i^{extd} = 0 && \wedge\\
(\alpha_{(i,i+1)}^{rest} = 1 \wedge \alpha_{i+1}^{rest} = 0) &\Rightarrow (l_{i+1}^{extd} = \mu_{(i,i+1)}^{extd3} + \mu_{i+1}^{extd}) && \wedge\\
(\alpha_{(i,i+1)}^{rest} = 0 \wedge \alpha_{i+1}^{rest} = 0) &\Rightarrow (l_{i+1}^{extd} = l_i^{extd} + \mu_{(i,i+1)}^{extd1} + \mu_{i+1}^{extd})
\end{aligned}$$

These conditions are represented by:

$$l_i^{extd} \le 1 - \alpha_i^{rest} \qquad \forall\ i = 0,\ldots,r-1 \tag{6.65}$$

$$l_{i+1}^{extd} \ge \mu_{i+1}^{extd} + \mu_{(i,i+1)}^{extd3} - \alpha_{i+1}^{rest} \qquad \forall\ i = 0,\ldots,r-2 \tag{6.66}$$

$$l_{i+1}^{extd} \le \mu_{i+1}^{extd} + \mu_{(i,i+1)}^{extd3} + 1 - \alpha_{(i,i+1)}^{rest} \qquad \forall\ i = 0,\ldots,r-2 \tag{6.67}$$

$$l_{i+1}^{extd} \ge l_i^{extd} + \mu_{(i,i+1)}^{extd1} + \mu_{i+1}^{extd} - \alpha_{(i,i+1)}^{rest} - \alpha_{i+1}^{rest} \qquad \forall\ i = 0,\ldots,r-2 \tag{6.68}$$

$$l_{i+1}^{extd} \le l_i^{extd} + \mu_{(i,i+1)}^{extd1} + \mu_{i+1}^{extd} + \alpha_{(i,i+1)}^{rest} \qquad \forall\ i = 0,\ldots,r-2 \tag{6.69}$$

Inequalities (6.66) and (6.67) are lower and upper bounds on l_{i+1}^{extd}, respectively. Therefore, $(1-\alpha_{(i,i+1)}^{rest})$ needs not to be subtracted in constraints (6.66) and α_{i+1}^{rest} needs not to be added in constraints (6.67). Similarly, α_{i+1}^{rest} needs not to be added in constraints (6.69).

The number of extended driving times during the week is bounded from above by the maximum between 0 and 2 minus the number of extended driving times that were already used since the start of the week.[1]

$$\sum_{i=0}^{r-2} \mu_{(i,i+1)}^{extd1} + \mu_{(i,i+1)}^{extd2} + \mu_{(i,i+1)}^{extd3} + \sum_{i=0}^{r-1} \mu_i^{extd} \le \max\{2 - \overline{noExt} - \overline{dte}, 0\} \tag{6.70}$$

As mentioned at the beginning, every daily driving time extension is coupled with a break, meaning that a break has to be scheduled for each driving time extension. Hence, the following constraints have to be added:

$$A_{(i,i+1)}^{break} \ge \mu_{(i,i+1)}^{extd1} + \mu_{(i,i+1)}^{extd2} + \mu_{(i,i+1)}^{extd3} \qquad \forall\ i = 0,\ldots,r-2 \tag{6.71}$$

$$\alpha_i^{break} + \alpha_i^{prest} - \mu_i^{prest} \ge \mu_i^{extd} \qquad \forall\ i = 0,\ldots,r-1 \tag{6.72}$$

[1] The extended driving times already used since the start of the week consist of extended driving times already completed with the start of a daily rest period ($\overline{noExt}$) and of a potential driving time extension still active when the schedule starts (if $\overline{dte} = 1$).

6.5 Begin of service constraints

Constraints (6.73) state that service, i.e. loading and/or unloading of goods, starting at time $start_{i+1}$ in vertex $i+1$ will exactly begin after the end of the preceding service time in vertex i ($\bar{\Delta}_i^{service}$) plus the driving time needed to reach destination $i+1$ from i ($\bar{\Delta}_{(i,i+1)}^{drive}$), plus the duration of all (partial) breaks and (partial) daily rest periods taken on arc $(i, i+1)$ and in vertex $i+1$, plus waiting time in $i+1$. If a first partial break was taken prior to the departure from i, 15 minutes are subtracted from the full duration of a corresponding subsequent break thus converting it to a second partial break with a duration of 30 minutes. If a break was substituted (for the substitution see page 32) by a first partial rest period, the 45 minutes for the break are subtracted.[1]

$$\begin{aligned} start_{i+1} = start_i &+ \bar{\Delta}_i^{service} + \bar{\Delta}_{(i,i+1)}^{drive} + 45\ A_{(i,i+1)}^{break} + \Delta_{(i,i+1)}^{rest} + 45\ \alpha_{i+1}^{break} + \Delta_{i+1}^{rest} \\ &+ \Delta_{i+1}^{wait} + 15\ \alpha_{i+1}^{pbreak} - 15\ \mu_{(i,i+1)}^{upbreak} - 15\ \mu_{i+1}^{upbreak} + 180\ \alpha_{i+1}^{prest} - 45\ \mu_{i+1}^{prest} \\ &\forall\ i = 0, \ldots, r-2 \end{aligned} \tag{6.73}$$

Vertex 0 denotes the starting position. (Partial) daily rest periods and 45 minute breaks are allowed in the first vertex and the continuation of already started (partial) daily rest periods and breaks is also considered. If a (partial) break (or daily rest) takes place at the starting time of the schedule, parameter $\overline{ubt}$ (or $\overline{urt}$, respectively) specifies its duration until then. The continuation of a (partial) break or daily rest period is not mandatory. The service time $\bar{\Delta}_0^{service}$ is set to be zero, that means, in (6.74) $start_0$ denotes the start of driving from vertex 0 to vertex 1. The utilization of a first partial break or a first partial daily rest period still active is also taken into consideration, where $\overline{hpb}$ indicates if a first partial break has already been taken. If a first partial daily rest period is still active, the duration of the corresponding daily rest period is adjusted accordingly to obtain a second partial daily rest period (see Section 6.9 on page 43).

$$\begin{aligned} start_0 = \Delta_0^{rest} &+ (45 - \min(\overline{ubt} + 15 \cdot \overline{hpb}, 45)) \cdot \alpha_0^{break} \\ &+ (15 - \min(\overline{ubt}, 15)) \cdot \alpha_0^{pbreak} + (180 - \min(\overline{urt}, 180)) \cdot \alpha_0^{prest} \end{aligned} \tag{6.74}$$

6.6 Time window constraints

We model time windows as soft constraints, i.e we penalize lateness that refers to a chosen time window in the objective function. Thus, a solution can be found even if not all time windows can be met giving additional helpful information to the dispatcher. To guarantee that exactly one time window is chosen for each vertex $i = 1, \ldots, r-1$, constraints (6.75)

[1] Note that first partial breaks and first partial daily rest periods are scheduled in vertices to compensate waiting time. Similar to breaks, first partial daily rest periods also reset the driving time interval and it may be advantageous to substitute a last break on an arc with a first partial daily rest period. In that case, $\mu_i^{prest} = 1$ and $\alpha_{i+1}^{prest} = 1$.

are introduced. Constraints (6.76) state that service in vertex i will start no earlier than the lower bound of the chosen time window.

$$\sum_{z=0}^{nTW_i-1} tw_{iz} = 1 \qquad \forall i = 1, \ldots, r-1 \tag{6.75}$$

$$start_i \geq \sum_{z=0}^{nTW_i-1} \overline{TW}_{iz}^{begin} tw_{iz} \qquad \forall i = 1, \ldots, r-1 \tag{6.76}$$

6.7 Lateness constraints

Lateness in vertex i is greater than or equal to the difference between the start of loading/unloading of goods and the end of the chosen time window (see (6.77)). The lateness variable Δ_i^{late} is defined to be greater than or equal to zero. Therefore, $\sum_{i=1}^{r-1} \Delta_i^{late}$ represents the total lateness that is penalized in the objective function.

$$\Delta_i^{late} \geq start_i - \sum_{z=0}^{nTW_i-1} \overline{TW}_{iz}^{end} tw_{iz} \qquad \forall i = 1, \ldots, r-1 \tag{6.77}$$

6.8 Maximum time between two consecutive weekly rest periods

The time between the end of a weekly rest period and the start of the following weekly rest period is not allowed to exceed 144 hours (8640 minutes). Therefore, each loading or unloading activity at a customer location has to end within this time interval:

$$start_i + \bar{\Delta}_i^{service} \leq 8640 - \overline{ptwr} \quad \forall\ i = 1, \ldots, r-1 \tag{6.78}$$

where $\overline{ptwr}$ denotes the time passed since the last weekly rest period at the beginning of the planning horizon in minutes.

6.9 Durations of daily rest periods

Depending on the number of daily rest periods $A_{(i,i+1)}^{rest}$ scheduled on an arc $(i, i+1)$, lower bounds on their cumulated duration $\Delta_{(i,i+1)}^{rest}$ are set up. The minimum cumulated duration is reduced by 2 hours for each reduced daily rest period on arc $(i, i+1)$ and for a second partial daily rest taken ($l_i^{prest} = 1$). The same applies to vertices but with the difference that at most one daily rest period per vertex may be scheduled. Note that if a daily rest period has been taken on arc $(i, i+1)$, a potential first partial daily rest period that has been taken in vertex i or in a vertex prior to i is exhausted and may not be used in vertex $i+1$. In vertex 0, we additionally have to consider the duration of an already started rest period ($\overline{urt} > 0$) and potentially a first partial daily rest period. We add the following constraints to our model accordingly:

$$\Delta_0^{rest} \geq \big(660 - \min(660, \overline{urt})\big)\ \alpha_0^{rest} - 120\ \mu_0^{redrest} - 120\ \overline{hpr} \tag{6.79}$$

$$\Delta_i^{rest} \geq 660\ \alpha_i^{rest} - 120\ \mu_i^{redrest} - 120\ l_{i-1}^{prest} \qquad \forall\ i = 1, \dots, r-1 \tag{6.80}$$

$$\Delta_{i+1}^{rest} \geq 660\ \alpha_{i+1}^{rest} - 120\ \mu_{i+1}^{redrest} - 120\ (1 - \alpha_{(i,i+1)}^{rest}) \qquad \forall\ i = 0, \dots, r-2 \tag{6.81}$$

$$\Delta_{(i,i+1)}^{rest} \geq 660\ A_{(i,i+1)}^{rest} - 120\ \mu_{(i,i+1)}^{redrest} - 120\ l_i^{prest} \qquad \forall\ i = 0, \dots, r-2 \tag{6.82}$$

If no daily rest is taken, we set the corresponding duration variable Δ_i^{rest} or $\Delta_{(i,i+1)}^{rest}$, respectively to be equal to zero:

$$\alpha_i^{rest} = 0 \Rightarrow \Delta_i^{rest} = 0 \qquad \wedge$$
$$\alpha_{(i,i+1)}^{rest} = 0 \Rightarrow \Delta_{(i,i+1)}^{rest} = 0$$

As an upper bound on the duration of a rest period, we choose the maximum time available between two weekly rest periods, i.e. 6·24 h = 144 h = 8640 min and thus obtain constraints (6.83) and (6.84).

$$\Delta_i^{rest} \leq 8640\ \alpha_i^{rest} \qquad \forall\ i = 0, \dots, r-1 \tag{6.83}$$

$$\Delta_{(i,i+1)}^{rest} \leq 8640\ A_{(i,i+1)}^{rest} \qquad \forall\ i = 0, \dots, r-2 \tag{6.84}$$

In this context, it may be advantageous to take daily rest periods with a duration of more than 11 hours or 9 hours, respectively. This influences the beginning of the next 24 h time interval in which a daily rest period has to be taken and may be necessary to cope with subsequent time windows.

6.10 Indicator variables for daily rests on arcs

The variable $\alpha_{(i,i+1)}^{rest}$ is used to indicate if at least one daily rest period is taken on arc $(i, i+1)$. Hence, the following condition has to hold for each $\alpha_{(i,i+1)}^{rest}$:

$$A_{(i,i+1)}^{rest} > 0 \Leftrightarrow \alpha_{(i,i+1)}^{rest} = 1$$

As the number of daily rests during one week is bounded from above by 8640:540=16[1], constraints (6.85) and (6.86) induce these conditions.[2]

$$16\ \alpha^{rest}_{(i,i+1)} \geq A^{rest}_{(i,i+1)} \qquad \forall\ i = 0, \dots, r-2 \tag{6.85}$$
$$\alpha^{rest}_{(i,i+1)} \leq A^{rest}_{(i,i+1)} \qquad \forall\ i = 0, \dots, r-2 \tag{6.86}$$

6.11 Indicator variables for breaks on arcs

In the following, some conditions will only hold if no breaks are taken or if at least one break is taken on an arc. Therefore, variable $\alpha^{break}_{(i,i+1)}$ is introduced to indicate if at least one break is taken on arc $(i, i+1)$ or not. We wish to state that

$$A^{break}_{(i,i+1)} > 0 \Leftrightarrow \alpha^{break}_{(i,i+1)} = 1.$$

An upper bound on the maximum number of breaks during one week is 8640 : 45 = 192. Again, using a big-M approach, the above statement is expressed by (6.87) and (6.88).

$$192\ \alpha^{break}_{(i,i+1)} \geq A^{break}_{(i,i+1)} \qquad \forall\ i = 0, \dots, r-2 \tag{6.87}$$
$$\alpha^{break}_{(i,i+1)} \leq A^{break}_{(i,i+1)} \qquad \forall\ i = 0, \dots, r-2 \tag{6.88}$$

6.12 Decision variables that indicate a necessary break

The decision variable l^{bn}_i indicates whether a break is necessary to completely use the daily driving time left when leaving vertex i (L^{ddt}_i) or not. This is the case if $L^{ddt}_i > L^{dt}_i$, where L^{dt}_i denotes the driving time left until the next break:

$$L^{ddt}_i > L^{dt}_i \Leftrightarrow l^{bn}_i = 1$$

This is imposed by the following constraints:

$$270\ l^{bn}_i \geq L^{ddt}_i - L^{dt}_i \qquad \forall\ i = 0, \dots, r-1 \tag{6.89}$$
$$l^{bn}_i \leq L^{ddt}_i - L^{dt}_i \qquad \forall\ i = 0, \dots, r-1 \tag{6.90}$$

Note that (6.90) induces $L^{ddt}_i \geq L^{dt}_i$ and (6.89) ensures that the maximum difference between L^{ddt}_i and L^{dt}_i is less than or equal to 270 min = 4.5 h. $L^{ddt}_i - L^{dt}_i$ is always either equal to zero, or greater than or equal to one, non-integer values between 0 and 1 are not possible due to the above constraints.

[1] A daily rest period has a duration of at least 9 h = 540 min and the time between two weekly rest periods is at most $24 \cdot 6$ h = 144 h = 8640 min.

[2] For the derivation see (f1') and (f2') on page 19.

6.13 Decision variables that indicate that a break has already been taken

Variable e_i^{bt} should indicate if the daily driving time left when entering vertex i, E_i^{ddt}, is greater than the driving time left until the next break has to be taken, E_i^{dt}, i.e. we wish to achieve that

$$E_i^{ddt} > E_i^{dt} \Leftrightarrow e_i^{bt} = 0.$$

This represented by constraints (6.91) and (6.92).

$$270\ (1 - e_i^{bt}) \geq E_i^{ddt} - E_i^{dt} \qquad \forall\ i = 0, \ldots, r-1 \tag{6.91}$$

$$1 - e_i^{bt} \leq E_i^{ddt} - E_i^{dt} \qquad \forall\ i = 0, \ldots, r-1 \tag{6.92}$$

In case at least one daily rest was made on arc $(i, i+1)$, e_i^{bt} indicates whether a break was taken since the last daily rest period when arriving at vertex i. Similar to the last section, non-integer values for the difference of E_i^{ddt} and E_i^{dt} between 0 and 1 are not possible due to inequalities (6.91) and (6.92).

6.14 Indicator variables for early daily rests

Instead of taking a break after 4.5 hours of driving to completely use the daily driving time until the next daily rest L_i^{ddt}, it is also possible to prepone a daily rest period. This may be advantageous if when leaving vertex i the daily driving time left, L_i^{ddt}, is only slightly larger than L_i^{dt}, the driving time left until the next break or rest period. If for example $L_i^{ddt} = L_i^{dt} + 10$, a break of a duration of 45 minutes as first resting activity on $(i, i+1)$ would only allow driving for another 10 minutes. In such a case, it can be advantageous to directly take a daily rest and to save 45 minutes to reach subsequent customer locations earlier. In other situations, just saving 45 minutes for an additional break is the only possibility to meet a subsequent customer time window (see for example Figure 10). It is not meaningful to take more than two early daily rests on one arc, and if two early daily rests are made, to have a positive effect, one of them has to be the first daily rest on that arc. Therefore, we distinguish between two types that also may be combined:

- Type 1: The first break on arc $(i, i+1)$ is substituted by an early daily rest ($\mu_{(i,i+1)}^{earlydr1} = 1$).
- Type 2: The last break on arc $(i, i+1)$ is substituted by an early daily rest ($\mu_{(i,i+1)}^{earlydr2} = 1$) and there are at least two daily rests on arc $(i, i+1)$.

The following conditions have to hold if early daily rests of type 1 or type 2 are scheduled. First, if no daily rest is scheduled at all on the arc $(i, i+1)$, then an early daily rest does

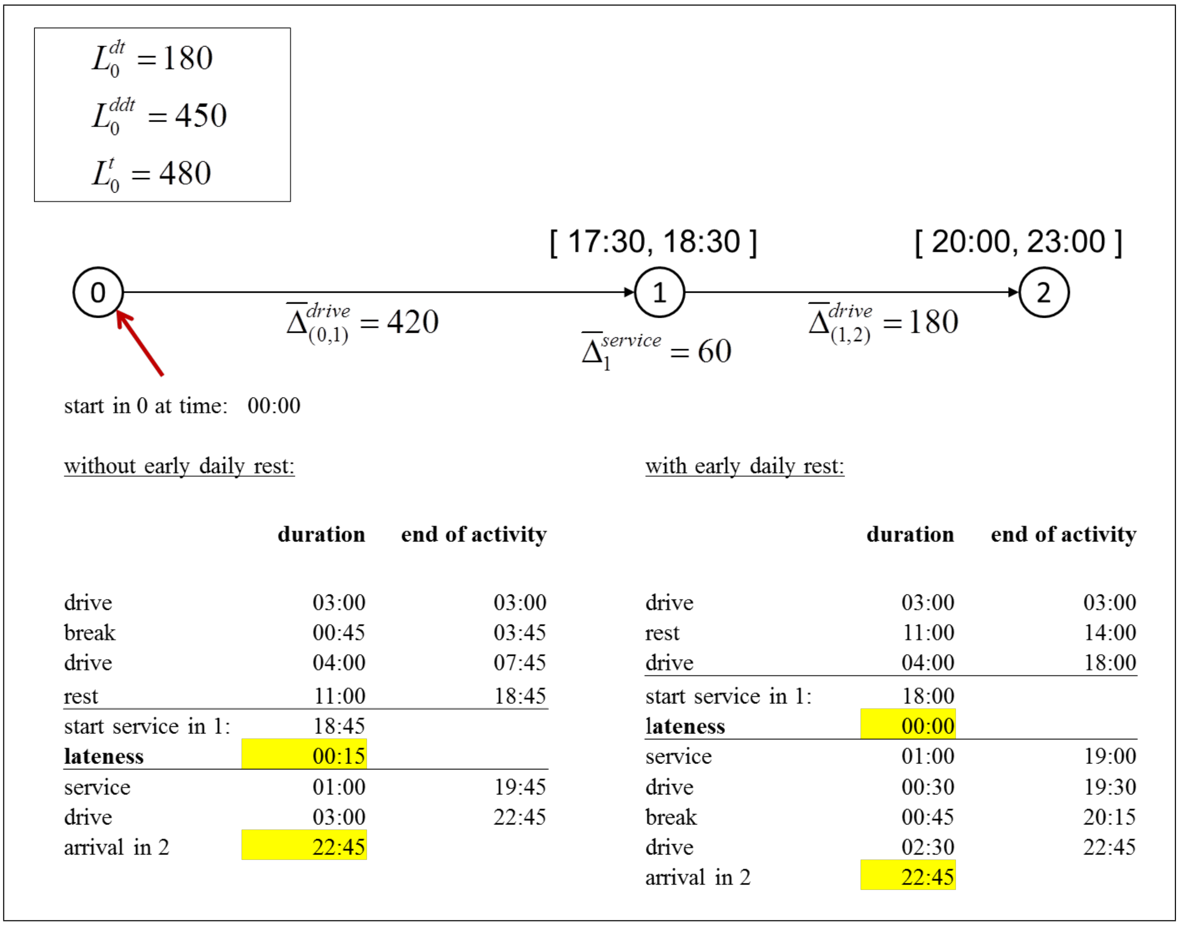

Figure 10: The impact of an early daily rest ($\mu_{(1,2)}^{earlydr2} = 1$)

not take place. If an early daily rest of type 2 is scheduled, there have to be at least two daily rest periods on the corresponding arc.

$$\alpha_{(i,i+1)}^{rest} = 0 \Rightarrow \mu_{(i,i+1)}^{earlydr1} = 0$$
$$A_{(i,i+1)}^{rest} < 2 \Rightarrow \mu_{(i,i+1)}^{earlydr2} = 0$$

The above conditions are expressed by the following constraints:

$$\mu_{(i,i+1)}^{earlydr1} \leq \alpha_{(i,i+1)}^{rest} \qquad \forall\ i = 0, \dots, r-2 \qquad (6.93)$$
$$\mu_{(i,i+1)}^{earlydr2} \leq A_{(i,i+1)}^{rest} - \alpha_{(i,i+1)}^{rest} \qquad \forall\ i = 0, \dots, r-2 \qquad (6.94)$$

If $L_i^{ddt} = L_i^{dt}$ then $l_i^{bn} = 0$ (because of (6.89) and (6.90)), this means that no break is necessary to completely use L_i^{ddt}. In this case, the subsequent daily rest period on arc $(i, i+1)$ (if there has to be one) has to be a regular daily rest. No early daily rest of type 1 can be scheduled. Therefore,

$$l_i^{bn} = 0 \Rightarrow \mu_{(i,i+1)}^{earlydr1} = 0$$

We obtain the following constraints:

$$\mu_{(i,i+1)}^{earlydr1} \leq l_i^{bn} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.95}$$

6.15 Vertex activity constraints

The vertex activity constraints generally limit possible activities and their combinations in a vertex.

In vertex 0, we decide to set waiting time to be zero (see (6.96)), as there is no time window:

$$\Delta_0^{wait} = 0 \tag{6.96}$$

Moreover, solutions with waiting time Δ_0^{wait} greater than zero would be equivalent (w.r.t. the objective function value) to solutions with waiting time in the following vertex in case no daily rest periods on arc $(0,1)$ were scheduled. If a daily rest period is taken on this arc, it may be extended accordingly.

If a break or daily rest period (full or partial) has not started yet, it may be postponed to the following arc $(0,1)$. Thus, to reduce the solution space, we add the constraints

$$\alpha_0^{break} = 0 \text{ and } \alpha_0^{pbreak} = 0 \text{ if } \overline{ubt} = 0 \tag{6.97}$$

$$\alpha_0^{rest} = 0 \text{ and } \alpha_0^{prest} = 0 \text{ if } \overline{urt} = 0 \tag{6.98}$$

and thereby prohibit starting a new rest period or break.

There is no clear rule concerning the time between a daily rest period and a partial break or partial rest. We assume that it is not desired by the legislator that these resting activities are scheduled directly in series and do only allow one resting activity per vertex:

$$\alpha_i^{rest} + \alpha_i^{break} + \alpha_i^{pbreak} + \alpha_i^{prest} - \mu_i^{prest} \leq 1 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.99}$$

If a break is substituted by a partial daily rest period ($\alpha_i^{prest} = 1 \ \wedge \ \mu_i^{prest} = 1$) on arc $(i-1, i)$, then a partial break for a driving time extension on a subsequent arc may be advantageous in contrast to the resting activities $\alpha_i^{break} = 1$ or $\alpha_i^{rest} = 1$. We add constraints (6.100) to further reduce the solution space.

$$\alpha_i^{rest} + \alpha_i^{break} + \alpha_i^{prest} \leq 1 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.100}$$

6.16 Get status constraints

The driver starts at vertex 0 with a certain status, which depends on former activities.[1] The time left until the next daily rest (E_0^t) depends on the time elapsed since the end of the last daily rest. The daily driving time left until the next daily rest (E_0^{ddt}) depends on the current daily driving time and the time left until the next daily rest period. The driving time left until the next break or daily rest period (E_0^{dt}) depends on the uninterrupted current driving time, the overall time spent driving since the last daily rest and the time left until the next daily rest. In addition, each of the status variables is influenced by a decision about a reduced daily rest period ($\mu_0^{dredrest} = 1$) and by a daily rest period scheduled in vertex 0.

The time left until the next daily rest has to be taken (E_0^t) depends on the time spent since the last daily rest period $\overline{ptr}$ and if a first partial rest was already made ($\overline{hpr} = 1$) in case no reduced daily rest period is taken in vertex 0.

As it may be the case that $\overline{ptr}$ is larger than actually allowed, $780 - \overline{ptr} + 120\ \overline{hpr}$ may be less than zero. The following logical condition takes this into account:

$$\mu_0^{dredrest} = 0 \Rightarrow E_0^t = \max\{780 - \overline{ptr} + 120\ \overline{hpr}, 0\}$$

For our model, we obtain constraints (6.101) and (6.102).

$$E_0^t \leq \max\{780 - \overline{ptr} + 120\ \overline{hpr}, 0\} + 120\ \mu_0^{dredrest} \tag{6.101}$$
$$E_0^t \geq \max\{780 - \overline{ptr} + 120\ \overline{hpr}, 0\} \tag{6.102}$$

If a reduced daily rest is planned after leaving vertex 0 ($\mu_0^{dredrest} = 1$), we have to differentiate between two cases. In case one, no daily rest is made in 0. In that case, two hours are added to E_0^t to be able to already use the additional time for activities in vertex 0. In case two, a daily rest period is taken in 0. In that case, two hours are added to L_0^t as the time left until the next rest period is reset by the daily rest period taken. The following two conditions must hold:

$$(\mu_0^{dredrest} = 1 \wedge \alpha_0^{rest} = 0) \Rightarrow E_0^t = \max\{900 - \overline{ptr}, 0\}$$
$$(\mu_0^{dredrest} = 1 \wedge \alpha_0^{rest} = 1) \Rightarrow E_0^t = \max\{780 - \overline{ptr}, 0\}$$

The following constraints enforce the above conditions:

$$E_0^t \leq \max\{780 - \overline{ptr} + 120\ \overline{hpr}, 0\} + 120\ (1 - \alpha_0^{rest}) \tag{6.103}$$
$$E_0^t \leq \max\{900 - \overline{ptr}, 0\} \tag{6.104}$$
$$E_0^t \geq \max\{900 - \overline{ptr}, 0\} - 120\ (1 - \mu_0^{dredrest}) - 120\ \alpha_0^{rest} \tag{6.105}$$

Note that a reduced daily rest period will not be taken if a second partial rest can be made instead (see constraints (6.46) on page 36).

[1] See page 14 for a short description of the status variables.

Now, the daily driving time left until the next daily rest (E_0^{ddt}) can be determined. E_0^{ddt} depends on the driving time already used since the last daily rest and a potential driving time extension. In addition, E_0^{ddt} is bounded from above by the time left until the next daily rest period E_0^{t}. We obtain the following conditions:

$$\begin{aligned}
&(\mu_0^{dredrest} = 0 \vee \ \alpha_0^{rest} = 1)\\
\Rightarrow &E_0^{ddt} = \max\{\min\{540 + 60\ \overline{dte} - \overline{ddt}, 780 - \overline{ptr} + 120\ \overline{hpr}\}, 0\}\\
&(\mu_0^{dredrest} = 1 \wedge \alpha_0^{rest} = 0)\\
\Rightarrow &E_0^{ddt} = \max\{\min\{540 + 60\ \overline{dte} - \overline{ddt}, 900 - \overline{ptr}\}, 0\}
\end{aligned}$$

By using the big-M approach and considering the fact that the different cases may only differ by at most 120 minutes as far as the decision about a reduced daily rest period $\mu_0^{dredrest}$ is concerned, we add the following constraints to our model:

$$\begin{aligned}
E_0^{ddt} \leq\ &\max\{\min\{540 + 60\ \overline{dte} - \overline{ddt}, 780 - \overline{ptr} + 120\ \overline{hpr}\}, 0\}\\
&+ 120\ \mu_0^{dredrest} && (6.106)\\
E_0^{ddt} \geq\ &\max\{\min\{540 + 60\ \overline{dte} - \overline{ddt}, 780 - \overline{ptr} + 120\ \overline{hpr}\}, 0\} && (6.107)\\
E_0^{ddt} \leq\ &\max\{\min\{540 + 60\ \overline{dte} - \overline{ddt}, 780 - \overline{ptr} + 120\ \overline{hpr}\}, 0\}\\
&+ 120\ (1 - \alpha_0^{rest}) && (6.108)\\
E_0^{ddt} \leq\ &\max\{\min\{540 + 60\ \overline{dte} - \overline{ddt}, 900 - \overline{ptr}, 0\} && (6.109)\\
E_0^{ddt} \geq\ &\max\{\min\{540 + 60\ \overline{dte} - \overline{ddt}, 900 - \overline{ptr}, 0\}\\
&- 120\ (1 - \mu_0^{dredrest}) - 120\ \alpha_0^{rest} && (6.110)
\end{aligned}$$

The last continuous status variable to be determined for vertex 0 is E_0^{dt}. In addition to the dependencies on E_0^{t} and E_0^{ddt}, the driving time since the last break or rest period, $\overline{udt}$, needs to be considered:

$$\begin{aligned}
&(\mu_0^{dredrest} = 0 \vee \alpha_0^{rest} = 1)\\
\Rightarrow &E_0^{dt} = \max\{\min\{270 - \overline{udt}, 540 + 60\ \overline{dte} - \overline{ddt}, 780 - \overline{ptr} + 120\ \overline{hpr}\}, 0\}\\
&(\mu_0^{dredrest} = 1 \wedge \alpha_0^{rest} = 0)\\
\Rightarrow &E_0^{dt} = \max\{\min\{270 - \overline{udt}, 540 + 60\ \overline{dte} - \overline{ddt}, 900 - \overline{ptr}\}, 0\}
\end{aligned}$$

We add the following constraints to our model:

$$E_0^{dt} \le \max\{\min\{270 - \overline{udt}, 540 + 60\ \overline{dte} - \overline{ddt}, 780 - \overline{ptr} + 120\ \overline{hpr}\}, 0\} + 120\ \mu_0^{dredrest} \tag{6.111}$$

$$E_0^{dt} \ge \max\{\min\{270 - \overline{udt}, 540 + 60\ \overline{dte} - \overline{ddt}, 780 - \overline{ptr} + 120\ \overline{hpr}\}, 0\} \tag{6.112}$$

$$E_0^{dt} \le \max\{\min\{270 - \overline{udt}, 540 + 60\ \overline{dte} - \overline{ddt}, 780 - \overline{ptr} + 120\ \overline{hpr}\}, 0\} + 120\ (1 - \alpha_0^{rest}) \tag{6.113}$$

$$E_0^{dt} \le \max\{\min\{270 - \overline{udt}, 540 + 60\ \overline{dte} - \overline{ddt}, 900 - \overline{ptr}\}, 0\} \tag{6.114}$$

$$E_0^{dt} \ge \max\{\min\{270 - \overline{udt}, 540 + 60\ \overline{dte} - \overline{ddt}, 900 - \overline{ptr}\}, 0\} - 120(1 - \mu_0^{dredrest}) - 120\ \alpha_0^{rest} \tag{6.115}$$

6.17 Continuous driver status variables when entering a vertex

The driver status variables when entering vertex $i+1$ (E_{i+1}^{dt}, E_{i+1}^{ddt} and E_{i+1}^{t}) are determined based on the driver status variables when leaving vertex i and the driver activities on arc $(i, i+1)$. Driver activities on arcs include resting, taking a break and driving. If a first partial daily rest or break has been taken in a preceding vertex, second partial rests and breaks can be scheduled. The interrelation of status variables and activities on arcs is shown in Figure 5 on page 13.

The constraints in this section serve two purposes. One is to determine the necessary number, duration and timing of rest periods and breaks for traversing an arc $(i, i+1)$ depending on the driver status L_i^{dt}, L_i^{ddt} and L_i^{t} when leaving i. The second purpose is the determination of the resulting driver status, E_i^{dt}, E_i^{ddt} and E_i^{t} when entering vertex $i+1$.

We introduce variable λ_i^5 to indicate if a driving time extension of one complete hour before the first daily rest period (if there is one) on arc $(i, i+1)$ is possible, without exceeding the time left until the next rest period (L_i^t). In case no driving time extension is considered on arc $(i, i+1)$ before the first daily rest period, λ_i^5 is set to be equal to 1.

$$(\mu_{(i,i+1)}^{extd1} = 1 \wedge L_i^t > L_i^{ddt} + 60 + 45\ l_i^{bn} + 45 - 15\ \mu_{(i,i+1)}^{upbreak}) \Rightarrow \lambda_i^5 = 1$$
$$(\mu_{(i,i+1)}^{extd1} = 1 \wedge L_i^t < L_i^{ddt} + 60 + 45\ l_i^{bn} + 45 - 15\ \mu_{(i,i+1)}^{upbreak}) \Rightarrow \lambda_i^5 = 0$$
$$\mu_{(i,i+1)}^{extd1} = 0 \Rightarrow \lambda_i^5 = 1$$

In case that $L_i^t = L_i^{ddt} + 60 + 45\ l_i^{bn} + 45 - 15\ \mu_{(i,i+1)}^{upbreak}$ and $\mu_{(i,i+1)}^{extd1} = 1$, λ_i^5 will not be uniquely determined. This causes no problem as the corresponding constraints in that case will yield the same variable values for E_i^{dt} or E_i^{ddt}, no matter if $\lambda_i^5 = 1$ or $\lambda_i^5 = 0$.[1] We

[1] See Section 6.1.2, page 18 for a more detailed description.

use the big-M approach to represent the above statements.

$$270\ \lambda_i^5 \geq L_i^t - L_i^{ddt} - 60 - 45\ l_i^{bn} - 45 + 15\ \mu_{(i,i+1)}^{upbreak} - 270\ (1 - \mu_{(i,i+1)}^{extd1}) \qquad \forall\ i = 0, \ldots, r-2 \tag{6.116}$$

$$150\ (\lambda_i^5 - 1) \leq L_i^t - L_i^{ddt} - 60 - 45\ l_i^{bn} - 45 + 15\ \mu_{(i,i+1)}^{upbreak} + 150\ (1 - \mu_{(i,i+1)}^{extd1}) \qquad \forall\ i = 0, \ldots, r-2 \tag{6.117}$$

$$\lambda_i^5 \geq 1 - \mu_{(i,i+1)}^{extd1} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.118}$$

Because of having a maximum difference of $900 - 540 = 360$ minutes for $L_i^t - L_i^{ddt}$, we set M to $360 - 60 - 45 + 15 = 270$ in inequalities (6.116). Knowing that $L_i^t \geq L_i^{ddt}$, we choose M to be $60 + 45 + 45 = 150$ in constraints (6.117).

6.17.1 Driving time left until the next break or rest

We will now determine E_{i+1}^{dt}, the driving time left until the next break or daily rest period when entering vertex i. The value of this variable depends on the driver status when leaving the preceding vertex i and the driver activities scheduled for arc $(i, i+1)$.

We have to differentiate between different cases when setting the value of E_{i+1}^{dt} depending on

- the driver status L_i^{dt} when leaving the previous vertex
- if an early daily rest period is taken as the first resting activity on arc $(i, i+1)$[1] ,
- the value of λ_i^5 and
- if a break is scheduled on the arc $(i, i+1)$.

Breaks and daily rest periods scheduled on the arc $(i, i+1)$ "extend" the driving time allowed until the next break or daily rest period starting with the value of the status variable E_i^{dt}. In most instances, each rest period or break allows 4.5 hours of additional driving to traverse the arc $(i, i+1)$. An exception can be the first resting activity on an arc if it is a break. The time left until the next daily rest period (L_i^t) may, due to waiting, loading and unloading activities in the past, not suffice to schedule another 4.5 hours of driving after this break. Another exception is a break that is taken for daily driving time extension which may extend the daily driving time by one hour (from 9 to a maximum of 10 hours). If such a break is made before the first daily rest period on that arc, it may be the case that the extension is less than 60 minutes as the maximum daily driving time is bounded from above by the maximum time between two consecutive daily rest periods.

Let us first determine the driving time left if no early daily rest is made as first resting activity ($\mu_{(i,i+1)}^{earlydr1} = 0$) and either a complete driving time extension of 60 minutes is possible before the first daily rest on arc $(i, i+1)$ or such a driving time extension is not

[1] Decision variables for early daily rests: see (6.93) - (6.94) on page 47.

made ($\lambda_i^5 = 1$). In addition, we demand that at least one break is necessary to traverse arc $(i, i+1)$.

$$
\begin{aligned}
&(\mu_{(i,i+1)}^{earlydr1} = 0 \wedge \lambda_i^5 = 1 \wedge \alpha_{(i,i+1)}^{break} = 1) \\
\Rightarrow &E_i^{dt} = L_i^{ddt} + 270\ A_{(i,i+1)}^{break} - 270\ l_i^{bn} + 270\ A_{(i,i+1)}^{rest} - \bar{\Delta}_{(i,i+1)}^{drive} \\
&\qquad - 210\ \mu_{(i,i+1)}^{extd1} - 210\ \mu_{(i,i+1)}^{extd2} - 210\ \mu_{(i,i+1)}^{extd3}
\end{aligned}
$$

If the first resting activity on arc $(i, i+1)$ is a break ($\alpha_{(i,i+1)}^{break} = 1 \wedge l_i^{bn} = 1 \wedge \mu_i^{earlydr1} = 0$), it may be the case that $L_i^{ddt} < L_i^{dt} + 270$, as L_i^{ddt} is bounded from above by the time left until the next daily rest L_i^t, which is influenced by former servicing and waiting times.[1] The first break then serves to completely use L_i^{ddt}, i.e. it extends the driving time by $L_i^{ddt} - L_i^{dt}$. Therefore, 270 minutes ($270\ l_i^{bn}$) need to be subtracted. In addition, 210 minutes need to be subtracted for each extended daily driving time, as the corresponding breaks only extend driving by 60, not by 270 minutes.

If a daily driving time extension ($\mu_{(i,i+1)}^{extd1} = 1$) is used and $\lambda_i^5 = 0$, then the remaining daily driving time depends on the time left until the next daily rest is necessary (L_i^t) minus 45 minutes for the break needed for the driving time extension minus 45 minutes if $L_i^{ddt} > L_i^{dt}$ ($l_i^{bn} = 1$) plus 15 minutes if a partial break has already been taken.

$$
\begin{aligned}
&\lambda_i^5 = 0 \\
\Rightarrow\ &E_{i+1}^{dt} = L_i^t + 270\ A_{(i,i+1)}^{break} - 270 - 45 - 270\ l_i^{bn} - 45\ l_i^{bn} + 15\ \mu_{(i,i+1)}^{upbreak} \\
&\qquad + 270\ A_{(i,i+1)}^{rest} - \bar{\Delta}_{(i,i+1)}^{drive} - 210\ \mu_{(i,i+1)}^{extd2} - 210\ \mu_{(i,i+1)}^{extd3}
\end{aligned}
$$

If an early daily rest period is scheduled (i.e. $\mu_{(i,i+1)}^{earlydr1} = 1$), the first resting activity on arc $(i, i+1)$ is a daily rest period which "extends" the driving time until the next break or rest period L_i^{dt} by 4.5 hours.

$$
\begin{aligned}
&\mu_{(i,i+1)}^{earlydr1} = 1 \\
\Rightarrow &E_{i+1}^{dt} = L_i^{dt} + 270\ A_{(i,i+1)}^{break} + 270\ A_{(i,i+1)}^{rest} - \bar{\Delta}_{(i,i+1)}^{drive} - 210\ \mu_{(i,i+1)}^{extd2} - 210\ \mu_{(i,i+1)}^{extd3}
\end{aligned}
$$

For the case that no break is made on arc $(i, i+1)$, we simply obtain the logical condition

$$
\alpha_{(i,i+1)}^{break} = 0 \Rightarrow E_{i+1}^{dt} = L_i^{dt} + 270\ A_{(i,i+1)}^{rest} - \bar{\Delta}_{(i,i+1)}^{drive}
$$

Reformulating the above conditions by using the big-M approach and integrating upper and lower bounds we obtain the following linear constraints:

$$
\begin{aligned}
E_{i+1}^{dt} \leq\ &L_i^{ddt} + 270\ A_{(i,i+1)}^{break} - 270\ l_i^{bn} + 270\ A_{(i,i+1)}^{rest} - \bar{\Delta}_{(i,i+1)}^{drive} \\
&- 210\ \mu_{(i,i+1)}^{extd1} - 210\ \mu_{(i,i+1)}^{extd2} - 210\ \mu_{(i,i+1)}^{extd3} \\
&+ 480\ \mu_{(i,i+1)}^{earlydr1} + 480\ (1 - \alpha_{(i,i+1)}^{break}) + 480\ (1 - \lambda_i^5)
\end{aligned}
$$

[1] Another reason may be that a break in a preceding vertex was taken after less than 4.5 hours of driving.

$$\forall\ i = 0, \ldots, r-2 \tag{6.119}$$

$$\begin{aligned}E^{dt}_{i+1} \geq\ & L^{ddt}_i + 270\ A^{break}_{(i,i+1)} - 270\ l^{bn}_i + 270\ A^{rest}_{(i,i+1)} - \bar{\Delta}^{drive}_{(i,i+1)} \\ & - 210\ \mu^{extd1}_{(i,i+1)} - 210\ \mu^{extd2}_{(i,i+1)} - 210\ \mu^{extd3}_{(i,i+1)} \\ & - 630\ (1 - \lambda^5_i) \\ & \forall\ i = 0, \ldots, r-2\end{aligned} \tag{6.120}$$

$$\begin{aligned}E^{dt}_{i+1} \leq\ & L^{t}_i + 270\ A^{break}_{(i,i+1)} - 270 - 45 - 270\ l^{bn}_i - 45\ l^{bn}_i + 15\ \mu^{upbreak}_{(i,i+1)} \\ & + 270\ A^{rest}_{(i,i+1)} - \bar{\Delta}^{drive}_{(i,i+1)} - 210\ \mu^{extd2}_{(i,i+1)} - 210\ \mu^{extd3}_{(i,i+1)} \\ & + 630\ \lambda^5_i \\ & \forall\ i = 0, \ldots, r-2\end{aligned} \tag{6.121}$$

$$\begin{aligned}E^{dt}_{i+1} \geq\ & L^{t}_i + 270\ A^{break}_{(i,i+1)} - 270 - 45 - 270\ l^{bn}_i - 45\ l^{bn}_i + 15\ \mu^{upbreak}_{(i,i+1)} \\ & + 270\ A^{rest}_{(i,i+1)} - \bar{\Delta}^{drive}_{(i,i+1)} - 210\ \mu^{extd2}_{(i,i+1)} - 210\ \mu^{extd3}_{(i,i+1)} \\ & - 270\ \lambda^5_i \\ & \forall\ i = 0, \ldots, r-2\end{aligned} \tag{6.122}$$

$$\begin{aligned}E^{dt}_{i+1} \leq\ & L^{dt}_i + 270\ A^{break}_{(i,i+1)} + 270\ A^{rest}_{(i,i+1)} - \bar{\Delta}^{drive}_{(i,i+1)} \\ & - 210\ \mu^{extd2}_{(i,i+1)} - 210\ \mu^{extd3}_{(i,i+1)} \\ & \forall\ i = 0, \ldots, r-2\end{aligned} \tag{6.123}$$

$$\begin{aligned}E^{dt}_{i+1} \geq\ & L^{dt}_i + 270\ A^{break}_{(i,i+1)} + 270\ A^{rest}_{(i,i+1)} - \bar{\Delta}^{drive}_{(i,i+1)} \\ & - 210\ \mu^{extd2}_{(i,i+1)} - 210\ \mu^{extd3}_{(i,i+1)} \\ & - 1110\ (1 - \mu^{earlydr1}_{(i,i+1)}) \\ & \forall\ i = 0, \ldots, r-2\end{aligned} \tag{6.124}$$

$$\begin{aligned}E^{dt}_{i+1} \geq\ & L^{dt}_i + 270\ A^{rest}_{(i,i+1)} - \bar{\Delta}^{drive}_{(i,i+1)} \\ & \forall\ i = 0, \ldots, r-2\end{aligned} \tag{6.125}$$

Constraints (6.123) set an upper bound for E^{dt}_{i+1}. Therefore, no big-M terms are needed for these constraints. Lower bounds on E^{dt}_{i+1} are imposed by constraints (6.120) in case $\lambda^5_i = 1$ and by constraints (6.122) if $\lambda^5_i = 0$. Accordingly, M is chosen for the remaining constraints such that they become redundant if the corresponding conditions do not hold. Often, the following dependencies are used:

- $L^{dt}_i \leq L^{ddt}_i \leq L^t_i \qquad \forall\ i = 0, \ldots, r-1$
- $L^t_i \leq L^{ddt}_i + 360 \qquad \forall\ i = 0, \ldots, r-1$

Big-M terms are always shown on the right-hand side of the constraint and appear at the end.

Finally, we demand that no break is made if it is not necessary. That means, if $L^{dt}_i > \bar{\Delta}^{drive}_{(i,i+1)}$, then no break is scheduled.

$$L^{dt}_i > \bar{\Delta}^{drive}_{(i,i+1)} \Rightarrow \alpha^{break}_{(i,i+1)} = 0$$

This is guaranteed by

$$270\ (1 - \alpha^{break}_{(i,i+1)}) \geq L^{dt}_i - \bar{\Delta}^{drive}_{(i,i+1)} \qquad \forall\ i = 0, \ldots, r-2 \tag{6.126}$$

6.17.2 Daily driving time left

Now, the driving time left until the next daily rest period when entering vertex i, E^{ddt}_i, will be determined. Similarly as in the last section, rest periods are considered to "extend" the driving time allowed until the next daily rest period starting with the value of the status variable E^{ddt}_i. Each daily rest period allows 9 hours of additional driving to traverse the arc $(i, i+1)$. Each extended daily driving time allows one additional hour of driving, except for the case that the daily driving time is extended before the first daily rest is taken (if one is taken) on arc $(i, i+1)$, as L^t_i, the time left until the next daily rest period when leaving vertex i may not suffice to schedule another 60 minutes of driving. An early daily rest as first resting activity ($\mu^{earlydr1}_{(i,i+1)} = 1$) reduces the driving time until the next daily rest period from L^{ddt}_i to L^{dt}_i, an early daily rest as last resting activity reduces the driving time between the last two daily rest periods on arc $(i, i+1)$ from 9 to 4.5 hours.

At first, let us set the logical condition for the case in which a driving time extension before the first daily rest period (if one is scheduled) is possible, without exceeding L^t_i ($\lambda^5_i = 1$) and no early daily rest period as first resting activity is scheduled ($\mu^{earlydr1}_{(i,i+1)} = 0$):

$$\begin{aligned}
&(\mu^{earlydr1}_{(i,i+1)} = 0 \wedge \lambda^5_i = 1)\\
\Rightarrow &E^{ddt}_{i+1} = L^{ddt}_i + 540\ A^{rest}_{(i,i+1)} - 270\ \mu^{earlydr2}_{(i,i+1)} - \bar{\Delta}^{drive}_{(i,i+1)}\\
&\qquad\quad + 60\ \mu^{extd1}_{(i,i+1)} + 60\ \mu^{extd2}_{(i,i+1)} + 60\ \mu^{extd3}_{(i,i+1)}
\end{aligned}$$

The next condition considers the case that a driving time extension is scheduled before the first daily rest period, but the time L^t_i does not suffice to schedule another 60 minutes of driving.[1]

$$\begin{aligned}
&\lambda^5_i = 0\\
\Rightarrow &E^{ddt}_{i+1} = L^t_i - 45\ l^{bn}_i - 45 + 15\ \mu^{upbreak}_{(i,i+1)} + 540\ A^{rest}_{(i,i+1)} - 270\ \mu^{earlydr2}_{(i,i+1)} - \bar{\Delta}^{drive}_{(i,i+1)}\\
&\qquad\quad + 60\ \mu^{extd2}_{(i,i+1)} + 60\ \mu^{extd3}_{(i,i+1)}
\end{aligned}$$

In case an early daily rest period is the first resting activity on arc $(i, i+1)$ ($\mu^{earlydr1}_{(i,i+1)} = 1$), a driving time extension before the first daily rest period is not possible. We obtain the logical condition

$$\begin{aligned}
&\mu^{earlydr1}_{(i,i+1)} = 1\\
\Rightarrow &E^{ddt}_{i+1} = L^{dt}_i + 540\ A^{rest}_{(i,i+1)} - 270\ \mu^{earlydr2}_{(i,i+1)} - \bar{\Delta}^{drive}_{(i,i+1)} + 60\ \mu^{extd2}_{(i,i+1)} + 60\ \mu^{extd3}_{(i,i+1)}
\end{aligned}$$

[1] Note that in this case $\mu^{earlydr1}_i = 0$ because of (6.118) on page 52 and (6.54) on page 39.

Again, we reformulate the above conditions by using the big-M approach, integrate upper and lower bounds, and obtain the following linear constraints:

$$\begin{aligned}
E_{i+1}^{ddt} \leq\ & L_i^{ddt} + 540\ A_{(i,i+1)}^{rest} - 270\ \mu_{(i,i+1)}^{earlydr2} - \bar{\Delta}_{(i,i+1)}^{drive} \\
& + 60\ \mu_{(i,i+1)}^{extd1} + 60\ \mu_{(i,i+1)}^{extd2} + 60\ \mu_{(i,i+1)}^{extd3} \\
& \forall\ i = 0, \ldots, r-2 \qquad (6.127) \\
E_{i+1}^{ddt} \geq\ & L_i^{ddt} + 540\ A_{(i,i+1)}^{rest} - 270\ \mu_{(i,i+1)}^{earlydr2} - \bar{\Delta}_{(i,i+1)}^{drive} \\
& + 60\ \mu_{(i,i+1)}^{extd1} + 60\ \mu_{(i,i+1)}^{extd2} + 60\ \mu_{(i,i+1)}^{extd3} \\
& - 330\ \mu_{(i,i+1)}^{earlydr1} - 330\ (1 - \lambda_i^5) \\
& \forall\ i = 0, \ldots, r-2 \qquad (6.128) \\
E_{i+1}^{ddt} \leq\ & L_i^{t} - 45\ l_i^{bn} - 45 + 15\ \mu_{(i,i+1)}^{upbreak} + 540\ A_{(i,i+1)}^{rest} - 270\ \mu_{(i,i+1)}^{earlydr2} - \bar{\Delta}_{(i,i+1)}^{drive} \\
& + 60\ \mu_{(i,i+1)}^{extd2} + 60\ \mu_{(i,i+1)}^{extd3} \\
& + 150\ \lambda_i^5 \\
& \forall\ i = 0, \ldots, r-2 \qquad (6.129) \\
E_{i+1}^{ddt} \geq\ & L_i^{t} - 45\ l_i^{bn} - 45 + 15\ \mu_{(i,i+1)}^{upbreak} + 540\ A_{(i,i+1)}^{rest} - 270\ \mu_{(i,i+1)}^{earlydr2} - \bar{\Delta}_{(i,i+1)}^{drive} \\
& + 60\ \mu_{(i,i+1)}^{extd2} + 60\ \mu_{(i,i+1)}^{extd3} \\
& - 600\ \lambda_i^5 \\
& \forall\ i = 0, \ldots, r-2 \qquad (6.130) \\
E_{i+1}^{ddt} \leq\ & L_i^{dt} + 540\ A_{(i,i+1)}^{rest} - 270\ \mu_{(i,i+1)}^{earlydr2} - \bar{\Delta}_{(i,i+1)}^{drive} \\
& + 60\ \mu_{(i,i+1)}^{extd2} + 60\ \mu_{(i,i+1)}^{extd3} \\
& + 330\ (1 - \mu_{(i,i+1)}^{earlydr1}) \\
& \forall\ i = 0, \ldots, r-2 \qquad (6.131) \\
E_{i+1}^{ddt} \geq\ & L_i^{dt} + 540\ A_{(i,i+1)}^{rest} - 270\ \mu_{(i,i+1)}^{earlydr2} - \bar{\Delta}_{(i,i+1)}^{drive} \\
& + 60\ \mu_{(i,i+1)}^{extd2} + 60\ \mu_{(i,i+1)}^{extd3} \\
& \forall\ i = 0, \ldots, r-2 \qquad (6.132)
\end{aligned}$$

Constraints (6.127) and (6.132) impose upper and lower bounds on E_i^{ddt}, respectively. Therefore, no big-M terms are needed for these constraints. Accordingly, M is chosen for the remaining constraints such that they become redundant when the corresponding conditions do not hold. The following bounds are used:

- $L_i^{dt} \leq L_i^{ddt} \leq L_i^{t} \qquad \forall\ i = 0, \ldots, r-1$
- $L_i^{t} \leq L_i^{dt} + 630 \qquad \forall\ i = 0, \ldots, r-1$
- $L_i^{ddt} \leq L_i^{dt} + 270 \qquad \forall\ i = 0, \ldots, r-1$ (see (6.89) on page 45)

Again, big-M terms always appear at the end of the corresponding inequalities. Additionally, we demand that the driving time left until the next daily rest period or break is less than or equal to the daily driving time left:

$$E_i^{dt} \leq E_i^{ddt} \qquad \forall\ i = 0, \ldots, r-1 \tag{6.133}$$

6.17.3 Maximum time until the next daily rest period

When traveling long distances, often, reaching the maximum daily driving time allowed forces a new daily rest period and the time between two daily rest periods is less than $24 - 11 = 13$ or $24 - 9 = 15$ hours, respectively.

The time left until the next daily rest period when entering vertex i (i.e. E_i^t) can be derived from the daily driving time left until the next rest period E_i^{ddt} in case at least one daily rest period was taken on arc $(i, i+1)$. Otherwise, E_i^t depends on the status when leaving the preceding vertex i and the overall duration of activities when traversing the arc $(i, i+1)$. In addition, we have to consider if a daily rest period is taken in vertex $i+1$. If the next daily rest period after leaving vertex $i+1$ is planned to be a reduced one ($l_{i+1}^{dredrest} = 1$) and a daily rest is made in $i+1$, this will only affect L_{i+1}^t, the time left until the next daily rest period when leaving vertex $i+1$. If no daily rest is made in $i+1$ and a daily rest period was taken on arc $(i, i+1)$, E_i^t needs to be modified in case that $\mu_{i+1}^{dredrest} = 1$.[1]

Let us first consider the case that at least one daily rest period is taken on arc $(i, i+1)$ and no daily rest period is scheduled for vertex $i+1$. The time left until the next daily rest period when entering vertex $i+1$ exceeds the daily driving time left until the next period by $13\,\text{h} - 9\,\text{h} = 240$ min, minus 45 minutes if a break has already been taken since the last daily rest period on arc $(i, i+1)$ ($e_{i+1}^{bt} = 1$) plus $2\,\text{h} = 120\,\text{min}$ if the next daily rest will be a reduced one ($l_{i+1}^{dredrest} = 1$) minus $60 + 45 = 105$ minutes if an extended daily driving time ($\mu_{(i,i+1)}^{extd3} = 1$) was scheduled since the last daily rest period.

$$\begin{aligned} &(\alpha_{(i,i+1)}^{rest} = 1 \wedge \alpha_{i+1}^{rest} = 0) \\ \Rightarrow &E_{i+1}^t = E_{i+1}^{ddt} + 240 - 45\ e_{i+1}^{bt} + 120\ l_{i+1}^{dredrest} - 105\ \mu_{(i,i+1)}^{extd3} \end{aligned}$$

In the case that a daily rest period is taken in vertex $i+1$, the decision about the next daily rest period being a reduced one only influences L_{i+1}^t but not E_{i+1}^t.

$$(\alpha_{(i,i+1)}^{rest} = 1 \wedge \alpha_{i+1}^{rest} = 1) \Rightarrow E_{i+1}^t = E_{i+1}^{ddt} + 240 - 45\ e_{i+1}^{bt} - 105\ \mu_{(i,i+1)}^{extd3}$$

If no daily rest period was taken on the arc $(i, i+1)$, we obtain

$$\begin{aligned} &\alpha_{(i,i+1)}^{rest} = 0 \\ \Rightarrow &E_{i+1}^t = L_i^t - \bar{\Delta}_{(i,i+1)}^{drive} - 45\ A_{(i,i+1)}^{break} + 15\ \mu_{(i,i+1)}^{upbreak} \end{aligned}$$

We add the following linear constraints to our model:

$$E_{i+1}^t \leq E_{i+1}^{ddt} + 240 - 45\ e_{i+1}^{bt} + 120\ l_{i+1}^{dredrest} - 105\ \mu_{(i,i+1)}^{extd3}$$

[1] Recall that by constraints (6.41) a decision about a reduced rest period $\mu_i^{dredrest}$ is linked with a daily rest period on the preceding arc or vertex.

$$\forall\ i = 0, \dots, r-2 \tag{6.134}$$

$$\begin{aligned} E^t_{i+1} \geq\ & E^{ddt}_{i+1} + 240 - 45\ e^{bt}_{i+1} + 120\ l^{dredrest}_{i+1} - 105\ \mu^{extd3}_{(i,i+1)} \\ & - 900\ \alpha^{rest}_{i+1} - 900\ (1 - \alpha^{rest}_{(i,i+1)}) \\ & \forall\ i = 0, \dots, r-2 \end{aligned} \tag{6.135}$$

$$\begin{aligned} E^t_{i+1} \leq\ & E^{ddt}_{i+1} + 240 - 45\ e^{bt}_{i+1} - 105\ \mu^{extd3}_{(i,i+1)} \\ & + 120\ (1 - \alpha^{rest}_{i+1}) + 120\ (1 - \alpha^{rest}_{(i,i+1)}) \\ & \forall\ i = 0, \dots, r-2 \end{aligned} \tag{6.136}$$

$$\begin{aligned} E^t_{i+1} \geq\ & E^{ddt}_{i+1} + 240 - 45\ e^{bt}_{i+1} - 105\ \mu^{extd3}_{(i,i+1)} \\ & - 780\ (1 - \alpha^{rest}_{(i,i+1)}) \\ & \forall\ i = 0, \dots, r-2 \end{aligned} \tag{6.137}$$

$$\begin{aligned} E^t_{i+1} \leq\ & L^t_i - \bar{\Delta}^{drive}_{(i,i+1)} - 45\ A^{break}_{(i,i+1)} + 15\ \mu^{upbreak}_{(i,i+1)} \\ & + (1200 + \frac{7}{6}\ \bar{\Delta}^{drive}_{(i,i+1)})\ \alpha^{rest}_{(i,i+1)} \\ & \forall\ i = 0, \dots, r-2 \end{aligned} \tag{6.138}$$

$$\begin{aligned} E^t_{i+1} \geq\ & L^t_i - \bar{\Delta}^{drive}_{(i,i+1)} - 45\ A^{break}_{(i,i+1)} + 15\ \mu^{upbreak}_{(i,i+1)} \\ & \forall\ i = 0, \dots, r-2 \end{aligned} \tag{6.139}$$

Constraints (6.134) and (6.139) impose upper and lower bounds on E^t_{i+1}, respectively. Additional lower and upper bounds that are used for determining appropriate big-M's are:

- $0 \leq E^{ddt}_{i+1} \leq 540$
- $0 \leq L^t_{i+1} \leq 900$

For determining an appropriate M for (6.138), we first calculate an upper bound for the number of breaks on arc $(i, i+1)$. Therefore, recall (6.120) on page 54:

$$\begin{aligned} E^{dt}_{i+1} \geq\ & L^{ddt}_i + 270\ A^{break}_{(i,i+1)} - 270\ l^{bn}_i + 270\ A^{rest}_{(i,i+1)} - \bar{\Delta}^{drive}_{(i,i+1)} \\ & - 210\ \mu^{extd1}_{(i,i+1)} - 210\ \mu^{extd2}_{(i,i+1)} - 210\ \mu^{extd3}_{(i,i+1)} - 630\ (1 - \lambda^5_i) \end{aligned}$$

$\Rightarrow$

$$\begin{aligned} 270\ A^{break}_{(i,i+1)} + 270\ A^{rest}_{(i,i+1)} \leq\ & E^{dt}_{i+1} + 270\ l^{bn} + \bar{\Delta}^{drive}_{(i,i+1)} + 210\ \mu^{extd1}_{(i,i+1)} + 210\ \mu^{extd2}_{(i,i+1)} \\ & + 210\ \mu^{extd3}_{(i,i+1)} + 630\ (1 - \lambda^5_i) \end{aligned}$$

$\Rightarrow$

$$270\ A^{break}_{(i,i+1)} \leq 2 \cdot 270 + \bar{\Delta}^{drive}_{(i,i+1)} + 3 \cdot 210 + 630$$

$\Leftrightarrow$

$$A^{break}_{(i,i+1)} \leq \frac{1800 + \bar{\Delta}^{drive}_{(i,i+1)}}{270}$$

We set $M = 1200 + \frac{7}{6}\bar{\Delta}^{drive}_{(i,i+1)}$ and derive:

$$
\begin{aligned}
& M = 1200 + \frac{7}{6}\bar{\Delta}^{drive}_{(i,i+1)} \\
\Leftrightarrow & M = 900 + \bar{\Delta}^{drive}_{(i,i+1)} + 45 \cdot \frac{1800 + \bar{\Delta}^{drive}_{(i,i+1)}}{270} \\
\Rightarrow & M \geq E^t_{i+1} - L^t_i + \bar{\Delta}^{drive}_{(i,i+1)} + 45\ A^{break}_{(i,i+1)} \\
\Leftrightarrow & E^t_{i+1} \leq L^t_i - \bar{\Delta}^{drive}_{(i,i+1)} - 45\ A^{break}_{(i,i+1)} + M \\
\Rightarrow & E^t_{i+1} \leq L^t_i - \bar{\Delta}^{drive}_{(i,i+1)} - 45\ A^{break}_{(i,i+1)} + 15\ \mu^{upbreak}_{(i,i+1)} + M
\end{aligned}
$$

So, $M = 1200 + \frac{7}{6}\bar{\Delta}^{drive}_{(i,i+1)}$ is an appropriate choice, as constraints (6.138) become redundant in case daily rest periods are scheduled on the corresponding arcs $(i, i+1)$.

6.18 Continuous driver status variables when leaving a vertex

The driver status when leaving vertex i results from the driver status when entering vertex i and the driver activities in vertex i. Driver activities include resting (partially, full or reduced), taking a break (partially or full), waiting and loading/unloading goods. This interrelation is shown in Figure 4 on page 13.

In addition, in case a daily rest period is taken in vertex i, the decision about the next daily rest period being a reduced one has to be taken into account. The two status variables L^{ddt}_i for the daily driving time left and L^{dt}_i for the driving time left until the next break when leaving vertex i are bounded from above by L^t_i, the maximum time allowed until the next daily rest period when leaving vertex i. So we start with the discussion of the constraints concerning L^t_i.

Driver activities in vertices are scheduled applying the rule that a resting activity is finished first before waiting and afterward loading and/or unloading goods may start.

6.18.1 Time left until the next daily rest period

For determining L^t_i, the time left until the next daily rest period when leaving vertex i, we have to consider two cases:

- Case 1: A daily rest period is taken in vertex i.
- Case 2: No daily period is taken in vertex i.

In the first case, the time left until the next daily rest period equals 13 hours or 15 hours in case the next daily rest period is planned to be a reduced one, minus waiting time and minus the time needed for loading or unloading.

$$\alpha^{rest}_i = 1 \Rightarrow L^t_i = 780 + 120\ l^{dredrest}_i - \Delta^{wait}_i - \bar{\Delta}^{service}_i$$

In the second case we have to consider E_i^t, the status when entering vertex $i \neq 0$, and additionally all other activities that are possible in a vertex. Note that if a first partial daily rest period is made, $3 + 9 = 12$ hours have to be subtracted from the 24 hours time interval for the complete daily rest period to obtain the time for the other activities in this time interval.[1] As one additional hour is needed for the daily rest period if it is split, 60 minutes are subtracted. The opportunity of substituting[2] the last break on arc $(i-1, i)$ by a first partial daily rest period ($\mu_i^{prest} = 1$) is also taken into account. In that case, the first partial daily rest period migrates from vertex i to the arc $(i-1, i)$ and it substitutes a break.

$$\begin{aligned}\alpha_i^{rest} = 0 \Rightarrow L_i^t = E_i^t &- \bar{\Delta}_i^{service} - 45\ \alpha_i^{break} + 15\ \mu_i^{upbreak} - 15\ \alpha_i^{pbreak} \\ &- 60\ \alpha_i^{prest} + 45\ \mu_i^{prest} - \Delta_i^{wait}\end{aligned}$$

In vertex 0, a (partial) break or a (first partial) daily rest period may take place or may be continued:

$$\begin{aligned}\alpha_0^{rest} = 0 \Rightarrow L_0^t = E_0^t &- (45 - \min(45, \overline{ubt} + 15\ \overline{hpb}))\ \alpha_0^{break} \\ &- (15 - \min(15, \overline{ubt}))\ \alpha_0^{pbreak} - (60 - \min(60, \overline{urt}))\ \alpha_0^{prest}\end{aligned}$$

Reformulating the above conditions by using the big-M approach and integrating upper and lower bounds we obtain the following linear constraints:

$$L_i^t \leq 780 + 120\ l_i^{dredrest} - \Delta_i^{wait} - \bar{\Delta}_i^{service} \quad \forall\ i = 0, \ldots, r-1 \tag{6.140}$$

$$\begin{aligned}L_i^t \geq\ & 780 + 120\ l_i^{dredrest} - \Delta_i^{wait} - \bar{\Delta}_i^{service} \\ & - 900\ (1 - \alpha_i^{rest}) \\ & \forall\ i = 0, \ldots, r-1\end{aligned} \tag{6.141}$$

$$\begin{aligned}L_i^t \leq\ & E_i^t - \bar{\Delta}_i^{service} - 45\ \alpha_i^{break} + 15\ \mu_i^{upbreak} - 15\ \alpha_i^{pbreak} - 60\ \alpha_i^{prest} \\ & + 45\ \mu_i^{prest} - \Delta_i^{wait} \\ & + 1020\ \alpha_i^{rest} \\ & \forall\ i = 1, \ldots, r-1\end{aligned} \tag{6.142}$$

$$\begin{aligned}L_i^t \geq\ & E_i^t - \bar{\Delta}_i^{service} - 45\ \alpha_i^{break} + 15\ \mu_i^{upbreak} - 15\ \alpha_i^{pbreak} - 60\ \alpha_i^{prest} \\ & + 45\ \mu_i^{prest} - \Delta_i^{wait} \\ & \forall\ i = 1, \ldots, r-1\end{aligned} \tag{6.143}$$

$$\begin{aligned}L_0^t \leq\ & E_0^t - (45 - \min(45, \overline{ubt} + 15\ \overline{hpb}))\ \alpha_0^{break} - (15 - \min(15, \overline{ubt}))\ \alpha_0^{pbreak} \\ & - (60 - \min(60, \overline{urt}))\ \alpha_0^{prest} \\ & + 1020\ \alpha_0^{rest}\end{aligned} \tag{6.144}$$

$$\begin{aligned}L_0^t \geq\ & E_0^t - (45 - \min(45, \overline{ubt} + 15\ \overline{hpb}))\alpha_0^{break} - (15 - \min(15, \overline{ubt}))\ \alpha_0^{pbreak} \\ & - (60 - \min(60, \overline{urt}))\ \alpha_0^{prest}\end{aligned} \tag{6.145}$$

[1] See rule 6 and optional rule 2 on page 6.

[2] See also page 32.

Constraints (6.140), (6.143) and (6.145) impose lower and upper bounds on L_i^t. Additionally, $L_i^t \leq 900$ and $E_i^t \leq 900$ has to hold. Big-M's for the other constraints were determined accordingly.

6.18.2 Daily driving time left

A set of auxiliary decision variables is needed to determine the driving time left until the next daily rest period, L_i^{ddt}, that will now be introduced. We use the following bounds in conjunction with the big-M approach to set the corresponding constraints:

- $0 \leq L_i^t \leq 900$
- $0 \leq E_i^{ddt} \leq 540$
- $0 \leq E_i^{dt} \leq 270$

The auxiliary decision variable λ_i^1 indicates if there is not enough time to take a break on arc $(i, i+1)$ and completely use the daily driving time left (including a potential driving time extension), in case no daily rest period was made or to take a break and drive 540 min in total in case a daily rest was made. If $\lambda_i^1 = 0$ and no rest was taken, the time until the next daily rest period suffices to schedule an additional break on the next arc before the next daily rest period starts even if no such break is actually needed. Hence, the following conditions must hold:

$$
\begin{aligned}
(\alpha_i^{rest} = 0 \land L_i^t > E_i^{ddt} + 45 - 15\ l_i^{pbreak} + 60\ \mu_i^{extd}) \Rightarrow \lambda_i^1 = 0 \\
(\alpha_i^{rest} = 0 \land L_i^t < E_i^{ddt} + 45 - 15\ l_i^{pbreak} + 60\ \mu_i^{extd}) \Rightarrow \lambda_i^1 = 1 \\
(\alpha_i^{rest} = 1 \land L_i^t > 540 + 45) \Rightarrow \lambda_i^1 = 0 \\
(\alpha_i^{rest} = 1 \land L_i^t < 540 + 45) \Rightarrow \lambda_i^1 = 1
\end{aligned}
$$

This is induced by:

$$
\begin{aligned}
870\ (1 - \lambda_i^1) \geq L_i^t - E_i^{ddt} - 45 + 15\ l_i^{pbreak} - 60\ \mu_i^{extd} - 870\ \alpha_i^{rest} \\
\forall\ i = 0, \ldots, r-1 \qquad (6.146)
\end{aligned}
$$

$$
\begin{aligned}
-645\ \lambda_i^1 \leq L_i^t - E_i^{ddt} - 45 + 15\ l_i^{pbreak} - 60\ \mu_i^{extd} + 645\ \alpha_i^{rest} \\
\forall\ i = 0, \ldots, r-1 \qquad (6.147)
\end{aligned}
$$

$$
315\ (1 - \lambda_i^1) \geq L_i^t - 585 - 315\ (1 - \alpha_i^{rest}) \qquad \forall\ i = 0, \ldots, r-1 \qquad (6.148)
$$

$$
-585\ \lambda_i^1 \leq L_i^t - 585 + 585\ (1 - \alpha_i^{rest}) \qquad \forall\ i = 0, \ldots, r-1 \qquad (6.149)
$$

The auxiliary decision variable λ_i^2 indicates if it is possible to take a break after E_i^{dt} or 270 (in case a rest period, break or partial rest is scheduled in i) minutes of driving after leaving vertex i before taking the next daily rest period.

$$((\alpha_i^{rest} = 1 \lor \alpha_i^{break} = 1 \lor (\alpha_i^{prest} = 1 \land \mu_i^{prest} = 0)) \land L_i^t > 315) \Rightarrow \lambda_i^2 = 0$$
$$((\alpha_i^{rest} = 1 \lor \alpha_i^{break} = 1 \lor (\alpha_i^{prest} = 1 \land \mu_i^{prest} = 0)) \land L_i^t < 315) \Rightarrow \lambda_i^2 = 1$$
$$((\alpha_i^{rest} = 0 \land \alpha_i^{break} = 0 \land (\alpha_i^{prest} = 0 \lor \mu_i^{prest} = 1)) \land L_i^t > E_i^{dt} + 45 - 15\ l_i^{pbreak}) \Rightarrow \lambda_i^2 = 0$$
$$((\alpha_i^{rest} = 0 \land \alpha_i^{break} = 0 \land (\alpha_i^{prest} = 0 \lor \mu_i^{prest} = 1)) \land L_i^t < E_i^{dt} + 45 - 15\ l_i^{pbreak}) \Rightarrow \lambda_i^2 = 1$$

We add the following constraints to our model:

$$585\ (1 - \lambda_i^2) \geq L_i^t - 315 - 585\ (1 - \alpha_i^{rest} - \alpha_i^{break} - \alpha_i^{prest} + \mu_i^{prest}) \quad \forall\ i = 0, \ldots, r-1 \tag{6.150}$$
$$-315\ \lambda_i^2 \leq L_i^t - 315 + 315\ (1 - \alpha_i^{rest} - \alpha_i^{break} - \alpha_i^{prest} + \mu_i^{prest}) \quad \forall\ i = 0, \ldots, r-1 \tag{6.151}$$
$$870\ (1 - \lambda_i^2) \geq L_i^t - E_i^{dt} - 45 + 15\ l_i^{pbreak} - 870\ \alpha_i^{rest} - 870\ \alpha_i^{break} - 870\ \alpha_i^{prest} + 870\ \mu_i^{prest} \quad \forall\ i = 0, \ldots, r-1 \tag{6.152}$$
$$-315\ \lambda_i^2 \leq L_i^t - E_i^{dt} - 45 + 15\ l_i^{pbreak} + 315\ \alpha_i^{rest} + 315\ \alpha_i^{break} + 315\ \alpha_i^{prest} - 315\ \mu_i^{prest} \quad \forall\ i = 0, \ldots, r-1 \tag{6.153}$$

Note that only one resting activity per vertex is allowed if $\mu_i^{prest} = 0$. If $\mu_i^{prest} = 1$, a first partial break may still be taken (see constraints (6.99) and (6.100) on page 48).

The auxiliary decision variable λ_i^3 indicates if E_i^{dt} or 270 minutes (in case a resting activity was made in i) of driving until the next daily rest period are allowed due to L_i^t. Hence,

$$(\alpha_i^{rest} = 1 \lor \alpha_i^{break} = 1 \lor (\alpha_i^{prest} = 1 \land \mu_i^{prest} = 0) \land L_i^t > 270) \Rightarrow \lambda_i^3 = 0$$
$$(\alpha_i^{rest} = 1 \lor \alpha_i^{break} = 1 \lor (\alpha_i^{prest} = 1 \land \mu_i^{prest} = 0) \land L_i^t < 270) \Rightarrow \lambda_i^3 = 1$$
$$(\alpha_i^{rest} = 0 \land \alpha_i^{break} = 0 \land (\alpha_i^{prest} = 0 \lor \mu_i^{prest} = 1) \land L_i^t > E_i^{dt}) \Rightarrow \lambda_i^3 = 0$$
$$(\alpha_i^{rest} = 0 \land \alpha_i^{break} = 0 \land (\alpha_i^{prest} = 0 \lor \mu_i^{prest} = 1) \land L_i^t < E_i^{dt}) \Rightarrow \lambda_i^3 = 1$$

The above conditions are enforced by

$$630\ (1-\lambda_i^3) \geq L_i^t - 270 - 630\ (1-\alpha_i^{rest}-\alpha_i^{break}-\alpha_i^{prest}+\mu_i^{prest}) \quad \forall\ i=0,\dots,r-1 \tag{6.154}$$

$$-270\ \lambda_i^3 \leq L_i^t - 270 + 270\ (1-\alpha_i^{rest}-\alpha_i^{break}-\alpha_i^{prest}+\mu_i^{prest}) \quad \forall\ i=0,\dots,r-1 \tag{6.155}$$

$$900\ (1-\lambda_i^3) \geq L_i^t - E_i^{dt} - 900\ \alpha_i^{rest} - 900\ \alpha_i^{break} - 900\ \alpha_i^{prest} + 900\ \mu_i^{prest} \quad \forall\ i=0,\dots,r-1 \tag{6.156}$$

$$-270\ \lambda_i^3 \leq L_i^t - E_i^{dt} + 270\ \alpha_i^{rest} + 270\ \alpha_i^{break} + 270\ \alpha_i^{prest} - 270\ \mu_i^{prest} \quad \forall\ i=0,\dots,r-1 \tag{6.157}$$

Since

$$\lambda_i^3 = 1 \Rightarrow \lambda_i^2 = 1$$

must hold, we add the constraints

$$\lambda_i^2 \geq \lambda_i^3 \qquad \forall\ i=0,\dots,r-1 \tag{6.158}$$

The last auxiliary decision variable λ_i^6 is needed to check if in case that no daily rest period is taken in i, the time left until the next daily rest period when leaving vertex i, L_i^t, is larger than the daily driving time left when entering vertex i plus 60 minutes if a driving time extension is used ($\mu_i^{extd} = 1$). If a daily rest period is taken in i, λ_i^6 is set to be equal to 1.

$$(L_i^t > E_i^{ddt} + 60\ \mu_i^{extd} \wedge \alpha_i^{rest} = 0) \Rightarrow \lambda_i^6 = 0$$
$$(L_i^t < E_i^{ddt} + 60\ \mu_i^{extd} \wedge \alpha_i^{rest} = 0) \Rightarrow \lambda_i^6 = 1$$
$$\alpha_i^{rest} = 1 \Rightarrow \lambda_i^6 = 1$$

We add the constraints:

$$900\ (1-\lambda_i^6) \geq L_i^t - E_i^{ddt} - 60\ \mu_i^{extd} - 900\ \alpha_i^{rest} \qquad \forall\ i=0,\dots,r-1 \tag{6.159}$$

$$-540\ \lambda_i^6 \leq L_i^t - E_i^{ddt} - 60\ \mu_i^{extd} + 540\ \alpha_i^{rest} \qquad \forall\ i=0,\dots,r-1 \tag{6.160}$$

$$\lambda_i^6 \geq \alpha_i^{rest} \qquad \forall\ i=0,\dots,r-1 \tag{6.161}$$

Now, we can determine the driving time left until the next daily rest period, L_i^{ddt}, when leaving vertex i. First of all we consider the case that $\lambda_i^1 = 0$ and a daily rest is made in i. The following condition must be satisfied:

$$(\lambda_i^1 = 0 \wedge \alpha_i^{rest} = 1) \Rightarrow L_i^{ddt} = 540$$

As $\lambda_i^1 = 0$, the time until the next daily rest period suffices for the maximum daily driving time of 9 hours and a 45 minute break that has to be taken after at most 4.5 hours of uninterrupted driving. For the case that no daily rest is made in i and $\lambda_i^1 = 0$, we have to impose that

$$(\lambda_i^1 = 0 \wedge \alpha_i^{rest} = 0) \Rightarrow L_i^{ddt} = E_i^{ddt} + 60\ \mu_i^{extd}$$

Here, λ_i^1 indicates that there is enough time until the next daily rest left to completely use the daily driving time left when entering i, including a 45 minute break (no matter if needed or not) plus 60 minutes if a driving time extension is used.

If $\lambda_i^1 = 1$ and $\lambda_i^2 = 0$, there is enough time to schedule 270 minutes of driving and to take a break (second part or full) right after leaving vertex i if a resting activity that extends the driving time until the next break or daily rest takes place in i. If no such resting activity takes place, there is enough time to schedule E_i^{dt} minutes of driving and a break (second part or full). The time until the next rest period, L_i^t, does not exceed the time needed to completely use 9 hours driving time gained by taking a daily rest period in i or to use the daily driving time left when entering vertex i, respectively. In that case, the daily driving time left L_i^{ddt} when leaving i equals the time left until the next daily rest L_i^t when leaving i minus the time needed for a break (full or second part).

$$(\lambda_i^1 = 1 \wedge \lambda_i^2 = 0) \Rightarrow L_i^{ddt} = L_i^t - 45 + 15\ l_i^{pbreak}$$

If $\lambda_i^2 = 1$ and $\lambda_i^3 = 0$, we differentiate between the case with a resting activity that extends the driving time in vertex i and the case without. We will first take a look at the case with no resting activity in i. Because of $\lambda_i^2 = 1$, the time until the next daily rest does not suffice to schedule E_i^{ddt} minutes of driving until the next daily rest and to take a break. But as $\lambda_i^3 = 0$, it follows that $E_i^{dt} \leq L_i^t \leq E_i^{dt} + 45 - 15\ l_i^{pbreak}$. That means, E_i^{dt} minutes of driving are possible but a break to extend the driving time is not possible, as it would not fit in the time interval between the last and the subsequent daily rest period.

$$(\lambda_i^2 = 1 \wedge \lambda_i^3 = 0 \wedge \alpha_i^{rest} = 0 \wedge \alpha_i^{break} = 0 \wedge (\alpha_i^{prest} = 0 \vee \mu_i^{prest} = 1) \Rightarrow L_i^{ddt} = E_i^{dt}$$

If a resting activity that extends the driving time is made in vertex i, we differentiate between the case that $\lambda_i^6 = 1$ and $\lambda_i^6 = 0$. If $\lambda_i^6 = 1$, 270 minutes of driving until the next daily rest are possible, otherwise, E_i^{ddt} minutes are possible plus 60 minutes if the daily driving time is extended with a corresponding break in vertex i.

$$\begin{aligned}
&(\lambda_i^2 = 1 \wedge \lambda_i^3 = 0 \wedge (\alpha_i^{rest} = 1 \vee \alpha_i^{break} = 1 \vee (\alpha_i^{prest} = 1 \wedge \mu_i^{prest} = 0)) \wedge \lambda_i^6 = 1) \\
&\Rightarrow L_i^{ddt} = 270 \\
&(\lambda_i^2 = 1 \wedge \lambda_i^3 = 0 \wedge (\alpha_i^{rest} = 1 \vee \alpha_i^{break} = 1 \vee (\alpha_i^{prest} = 1 \wedge \mu_i^{prest} = 0)) \wedge \lambda_i^6 = 0) \\
&\Rightarrow L_i^{ddt} = E_i^{ddt} + 60\ \mu_i^{extd}
\end{aligned}$$

In case $\lambda_i^3 = 1$, the time left until the next daily rest does not suffice to drive 270 minutes, in case a resting activity that extends the driving time was made in i. Moreover, it does not suffice to drive E_i^{dt} minutes in case no resting activity was made. Depending on λ_i^6, we can determine L_i^{ddt} as follows:

$$\begin{aligned}
&(\lambda_i^3 = 1 \wedge \lambda_i^6 = 1) \Rightarrow L_i^{ddt} = L_i^t \\
&(\lambda_i^3 = 1 \wedge \lambda_i^6 = 0) \Rightarrow L_i^{ddt} = E_i^{ddt} + 60\ \mu_i^{extd}
\end{aligned}$$

Again, we use the big-M approach and integrate upper and lower bounds to derive linear constraints. Redundant constraints like $L_i^{ddt} \leq 540$ are omitted.

$$L_i^{ddt} \geq 540 - 540\ \lambda_i^1 - 540\ (1 - \alpha_i^{rest})$$

$$\forall\ i = 0, \ldots, r-1 \tag{6.162}$$

$$L_i^{ddt} \leq E_i^{ddt} + 60\ \mu_i^{extd} + 540\ \alpha_i^{rest}$$
$$\forall\ i = 0, \ldots, r-1 \tag{6.163}$$

$$L_i^{ddt} \geq E_i^{ddt} + 60\ \mu_i^{extd} - 600\ \alpha_i^{rest} - 600\ \lambda_i^1$$
$$\forall\ i = 0, \ldots, r-1 \tag{6.164}$$

$$L_i^{ddt} \leq L_i^t - 45 + 15\ l_i^{pbreak} + 45\ (1 - \lambda_i^1) + 45\ \lambda_i^2$$
$$\forall\ i = 0, \ldots, r-1 \tag{6.165}$$

$$L_i^{ddt} \geq L_i^t - 45 + 15\ l_i^{pbreak} - 855\ (1 - \lambda_i^1) - 855\ \lambda_i^2$$
$$\forall\ i = 0, \ldots, r-1 \tag{6.166}$$

$$L_i^{ddt} \leq E_i^{dt} + 540\ (1 - \lambda_i^2) + 540\ \lambda_i^3 + 540\ \alpha_i^{break} + 540\ \alpha_i^{rest} + 540\ \alpha_i^{prest}$$
$$- 540\ \mu_i^{prest}$$
$$\forall\ i = 0, \ldots, r-1 \tag{6.167}$$

$$L_i^{ddt} \geq E_i^{dt} - 270\ (1 - \lambda_i^2) - 270\ \lambda_i^3 - 270\ \alpha_i^{break} - 270\ \alpha_i^{rest} - 270\ \alpha_i^{prest}$$
$$+ 270\ \mu_i^{prest}$$
$$\forall\ i = 0, \ldots, r-1 \tag{6.168}$$

$$L_i^{ddt} \leq 270 + 270\ (1 - \lambda_i^2)$$
$$\forall\ i = 0, \ldots, r-1 \tag{6.169}$$

$$L_i^{ddt} \geq 270 - 270\ (1 - \lambda_i^2) - 270\ \lambda_i^3 - 270\ (1 - \alpha_i^{break} - \alpha_i^{rest} - \alpha_i^{prest} + \mu_i^{prest})$$
$$- 270\ (1 - \lambda_i^6)$$
$$\forall\ i = 0, \ldots, r-1 \tag{6.170}$$

$$L_i^{ddt} \leq E_i^{ddt} + 60\ \mu_i^{extd} + 540\ (1 - \lambda_i^2) + 540\ \lambda_i^3 +$$
$$540\ (1 - \alpha_i^{break} - \alpha_i^{prest} + \mu_i^{prest}) + 540\ \lambda_i^6$$
$$\forall\ i = 0, \ldots, r-1 \tag{6.171}$$

$$L_i^{ddt} \geq E_i^{ddt} + 60\ \mu_i^{extd} - 540\ (1 - \lambda_i^2) - 540\ \lambda_i^3$$
$$- 540\ (1 - \alpha_i^{break} - \alpha_i^{prest} + \mu_i^{prest}) - 540\ \lambda_i^6$$
$$\forall\ i = 0, \ldots, r-1 \tag{6.172}$$

$$L_i^{ddt} \leq L_i^t$$
$$\forall\ i = 0, \ldots, r-1 \tag{6.173}$$

$$L_i^{ddt} \geq L_i^t - 900\ (1 - \lambda_i^3) - 900\ (1 - \lambda_i^6)$$
$$\forall\ i = 0, \ldots, r-1 \tag{6.174}$$

$$L_i^{ddt} \geq E_i^{ddt} + 60\ \mu_i^{extd} - 540\ (1 - \lambda_i^3) - 540\ \lambda_i^6$$
$$\forall\ i = 0, \ldots, r-1 \tag{6.175}$$

6.18.3 Driving time left until the next break or daily rest period

Using L_i^{ddt} as an upper bound, the driving time left until the next break or daily rest period L_i^{dt} when leaving vertex i can now be easily determined.

If a resting activity was made in vertex i, L_i^{dt} depends on whether $L_i^{ddt} > 270$ or not. Therefore, we introduce the auxiliary variable λ_i^4:

$$L_i^{ddt} > 270 \Rightarrow \lambda_i^4 = 1$$
$$L_i^{ddt} < 270 \Rightarrow \lambda_i^4 = 0$$

We formulate this in our model as follows:

$$270\ \lambda_i^4 \geq L_i^{ddt} - 270 \qquad \forall\ i = 0, \dots, r-1 \qquad (6.176)$$
$$270\ (\lambda_i^4 - 1) \leq L_i^{ddt} - 270 \qquad \forall\ i = 0, \dots, r-1 \qquad (6.177)$$

With λ_i^4 we obtain the following conditions:

$$(\alpha_i^{rest} = 1 \vee \alpha_i^{break} = 1 \vee (\alpha_i^{prest} = 1 \wedge \mu_i^{prest} = 0) \wedge \lambda_i^4 = 1) \Rightarrow L_i^{dt} = 270$$
$$(\alpha_i^{rest} = 1 \vee \alpha_i^{break} = 1 \vee (\alpha_i^{prest} = 1 \wedge \mu_i^{prest} = 0) \wedge \lambda_i^4 = 0) \Rightarrow L_i^{dt} = L_i^{ddt}$$

In case no extending rest activity is made in vertex i, E_i^{dt} depends on whether $L_i^t > E_i^{dt}$ or not. In the last section, we introduced the variable λ_i^3 that is equal to zero if $L_i^t > E_i^{dt}$, and equal to one if $L_i^t < E_i^{dt}$ in case no extending rest activity is made. We obtain the following conditions:

$$(\alpha_i^{rest} = 0 \wedge \alpha_i^{break} = 0 \wedge (\alpha_i^{prest} = 0 \vee \mu_i^{prest} = 1) \wedge \lambda_i^3 = 1) \Rightarrow L_i^{dt} = L_i^{ddt}$$
$$(\alpha_i^{rest} = 0 \wedge \alpha_i^{break} = 0 \wedge (\alpha_i^{prest} = 0 \vee \mu_i^{prest} = 1) \wedge \lambda_i^3 = 0) \Rightarrow L_i^{dt} = E_i^{dt}$$

These logical conditions are induced by the following constraints:

$$L_i^{dt} \geq 270 - 270\ (1 - \lambda_i^4) - 270\ (1 - \alpha_i^{rest} - \alpha_i^{break} - \alpha_i^{prest} + \mu_i^{prest}) \qquad \forall\ i = 0, \dots, r-1 \qquad (6.178)$$

$$L_i^{dt} \leq L_i^{ddt} \qquad \forall\ i = 0, \dots, r-1 \qquad (6.179)$$

$$L_i^{dt} \geq L_i^{ddt} - 540\ \lambda_i^4 - 540\ (1 - \alpha_i^{rest} - \alpha_i^{break} - \alpha_i^{prest} + \mu_i^{prest}) \qquad \forall\ i = 0, \dots, r-1 \qquad (6.180)$$

$$L_i^{dt} \geq L_i^{ddt} - 540\ (1 - \lambda_i^3) - 540\ \alpha_i^{rest} - 540\ \alpha_i^{break} - 540\ \alpha_i^{prest} + 540\ \mu_i^{prest} \qquad \forall\ i = 0, \dots, r-1 \qquad (6.181)$$

$$L_i^{dt} \leq E_i^{dt} + 270\ \alpha_i^{rest} + 270\ \alpha_i^{break} + 270\ \alpha_i^{prest} - 270\ \mu_i^{prest} \qquad \forall\ i = 0, \dots, r-1 \qquad (6.182)$$

$$L_i^{dt} \geq E_i^{dt} - 270\ \lambda_i^3 \qquad \forall\ i = 0, \dots, r-1 \qquad (6.183)$$

6.19 Objective functions

The main objective is to minimize the sum of the overall lateness. The completion time, i.e. the overall schedule duration until the last customer is serviced and the last stop is reached is important, too. Other criteria are relevant for the quality of a solution in practice, but are considered less important. We chose a combination of strategies for this multicriteria optimization problem. At first, two sets were created. One for lateness and completion time and one for all other criteria. Within both sets a trade-off strategy was chosen providing different weights for each of the objectives. As the first set was considered to be more important than the second one, a lexicographic ordering was done for the two sets. Two objective functions were created from the two sets. The first one that is considered most important is described in Section 6.19.1 and the second one is described in Section 6.19.2.

6.19.1 Objective function 1

Penalized lateness for violating time windows is minimized along with the completion time.

$$\textbf{Minimize} \quad start_{r-1} + \sum_{i=1}^{r-1} P \cdot \Delta_i^{late} \tag{6.184}$$

with P as a user-specified penalty constant.

Note that the begin of service time $start_{r-1}$ is also penalized (with factor 1) in the objective function. As we consider lateness to be the most important optimization criterion, the penalty of 1 minute lateness is more important (i.e. has higher weight) than the penalty of the latest completion time possible. The maximum time between two weekly rest periods is $6 \cdot 24$ h $= 8640$ min. The schedule starts directly after finishing a weekly rest period (time 0). The latest completion time $start_{r-1} + \bar{\Delta}_{r-1}^{service}$ is therefore less than 8640. We set $P = 8640$ neglecting the time needed for loading and/or unloading at the last customer location, $\bar{\Delta}_{r-1}^{service}$.

6.19.2 Objective function 2

It is important that dispatchers and drivers agree with the schedules generated as otherwise they will not adopt them. Different criteria that are important for the quality of a solution and thus the compliance with the resulting schedule have not been taken into account until now. As dispatchers, drivers and anyone else that tries to analyze the schedule would be confused if the schedule contained, for example, unnecessary early daily rests or if waiting time would occur even though the lower boundary of the chosen time window had already been exceeded, such cases have to be eliminated to obtain more comprehensible solutions. Furthermore, it may be advantageous if the solution does not exploit driving time extensions or reduced daily rest periods completely. This is, for example, the case if the planning horizon does not comprise the whole week. Moreover, if unexpected events during the execution of the plan such as traffic jams or delays in loading or unloading do occur, using the optional rules may help to compensate the additional time needed. So the idea is to only use optional rules if there is any benefit. In other words, not making use of the optional rules would worsen lateness and/or completion time.

The objective function (6.185) contains different, partially conflicting, criteria needed to meet the objectives described above. They are provided with different weights that may be customized.

$$\begin{aligned}
\textbf{Minimize} \quad & \sum_{i=1}^{r-1} \sum_{z=0}^{nTW} 10\ (z + r - i)\ tw_{iz} + \sum_{i=0}^{r-1} start_i \\
& + \sum_{i=0}^{r-2} 10\ (r-i)\ (\mu_{(i,i+1)}^{earlydr1} + \mu_{(i,i+1)}^{earlydr2}) \\
& + \sum_{i=0}^{r-1} 10\ (r-i)\ (\alpha_i^{pbreak} + \alpha_i^{prest}) \\
& + \sum_{i=0}^{r-1} 20\ \Delta_i^{wait} \\
& + \sum_{i=0}^{r-2} 30\ (r-i)\ \mu_{(i,i+1)}^{redrest} + \sum_{i=0}^{r-1} 40\ (r-i)\ \mu_i^{redrest} \\
& + \sum_{i=0}^{r-2} 50\ (r-i)\ \mu_{(i,i+1)}^{extd2} + 60\ (r-i)\ \mu_{(i,i+1)}^{extd1} + 60\ (r-i)\ \mu_{(i,i+1)}^{extd3} \\
& + \sum_{i=0}^{r-1} 60\ (r-i)\ \mu_i^{extd} \qquad (6.185)
\end{aligned}$$

Note that penalizing waiting time has the positive side effect that daily rest periods are extended to compensate for waiting time.

6.20 The lexicographic solution technique

The constraints described in the previous sections together with the two objective functions presented in Section 6.19 form the MILP model with the consideration of optional rules. Two MILP submodels are created to be able to solve the problem in two optimization steps. The objective function (6.184) together with the constraints described form one submodel. This submodel is solved in the first optimization step. In the second optimization step, the objective function of the previous step is transformed to constraint (6.186) with the previous objective function value z^* as an upper bound such that the weighted sum of overall lateness and completion time is prevented from increasing.

$$start_{r-1} + \sum_{i=1}^{r-1} P \cdot \Delta_i^{late} \leq z^* \tag{6.186}$$

Together with the objective function (6.185) and the constraints from the previous optimization step we obtain the second submodel which is solved in the last optimization step. The solution obtained is the solution of the complete MILP model.

6.21 The MILP model without optional rules

If no optional rules should be taken into account, the MILP model can easily be adapted. By adding the following constraints, all optional rules are switched off.

$$\mu_{(i,i+1)}^{redrest} = 0 \qquad \forall\ i = 0, \ldots, r-2 \tag{6.187}$$
$$\mu_{(i,i+1)}^{extd1} = 0 \qquad \forall\ i = 0, \ldots, r-2 \tag{6.188}$$
$$\mu_{(i,i+1)}^{extd2} = 0 \qquad \forall\ i = 0, \ldots, r-2 \tag{6.189}$$
$$\mu_{(i,i+1)}^{extd3} = 0 \qquad \forall\ i = 0, \ldots, r-2 \tag{6.190}$$
$$\mu_{(i,i+1)}^{upbreak} = 0 \qquad \forall\ i = 0, \ldots, r-2 \tag{6.191}$$
$$\mu_{(i,i+1)}^{prest} = 0 \qquad \forall\ i = 0, \ldots, r-2 \tag{6.192}$$
$$\mu_i^{redrest} = 0 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.193}$$
$$\alpha_i^{prest} = 0 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.194}$$
$$\alpha_i^{pbreak} = 0 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.195}$$
$$\mu_i^{extd} = 0 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.196}$$
$$\mu_i^{dredrest} = 0 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.197}$$
$$\mu_i^{upbreak} = 0 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.198}$$
$$l_i^{pbreak} = 0 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.199}$$
$$l_i^{prest} = 0 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.200}$$
$$l_i^{dredrest} = 0 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.201}$$
$$l_i^{extd} = 0 \qquad \forall\ i = 0, \ldots, r-1 \tag{6.202}$$

These constraints are needed later when the influence of the optional rules on the run time and the objective function value is studied. Additionally, we will later use the solution of the MILP model with disabled optional rules as an upper bound for the optimal objective function value of the model with optional rules (only considering objective function (6.184)).

7 Numerical experiments - Part 1

The MILP models described in the previous section were implemented in Java (Java 8, 64 bit) and were solved with CPLEX 12.6 (64 bit) with ILOG Cplex Concert Technology. The test runs were performed on an Intel Core i5 2500K with 8 GB RAM (DDR3-10700 (667 MHz)) running Windows 7 Professional Service Pack 1, 64 bit.

If the number of time windows per customer location (vertices 1 to $r-2$) is denoted by z and the final destination is equipped with a "time window" that equals the planning horizon, the first MILP submodel (for minimizing lateness along with completion time) with optional rules has

- $28r + rz - 2z - 7$ binary variables,
- $4(r-1)$ integer (non-binary) variables,
- $11r - 2$ continuous variables and
- $147r - 49$ constraints

where r denotes the number of customer locations to be visited (including start and end position).

In the following, test instances are derived from real routes. Then the run times for two possible solution processes for the model with optional rules are compared. If no specific optimization step is addressed, the run times comprise the CPLEX times for all optimization steps involved and are given as wall clock time[1]. Afterwards, run times are analyzed depending on the number of stops and the number and size of time windows. The section concludes with the analysis of the influence of the optional rules on the run time as well as on lateness and overall traveling time.

7.1 Test instances

Test instances were derived from real data provided by a German haulage company that operates a fleet of vehicles in Europe. The haulage company was a partner in the research project Dynaserv which aimed at supporting dynamic tasks of dispatchers in transport companies by the integration of online data. The underlying database of the prototype developed during the project comprised telematics data of the vehicles of the haulage company as well as arrival times at customer locations initially planned by the dispatchers

[1] wall clock time: total physical time elapsed

in the order management system. Routes and driving times between customer locations were first calculated using the routing algorithm of the prototype. For each route, support points[1] were manually added until the route computed by the routing algorithm matched the route chosen by the driver. Figure 11 shows an example.

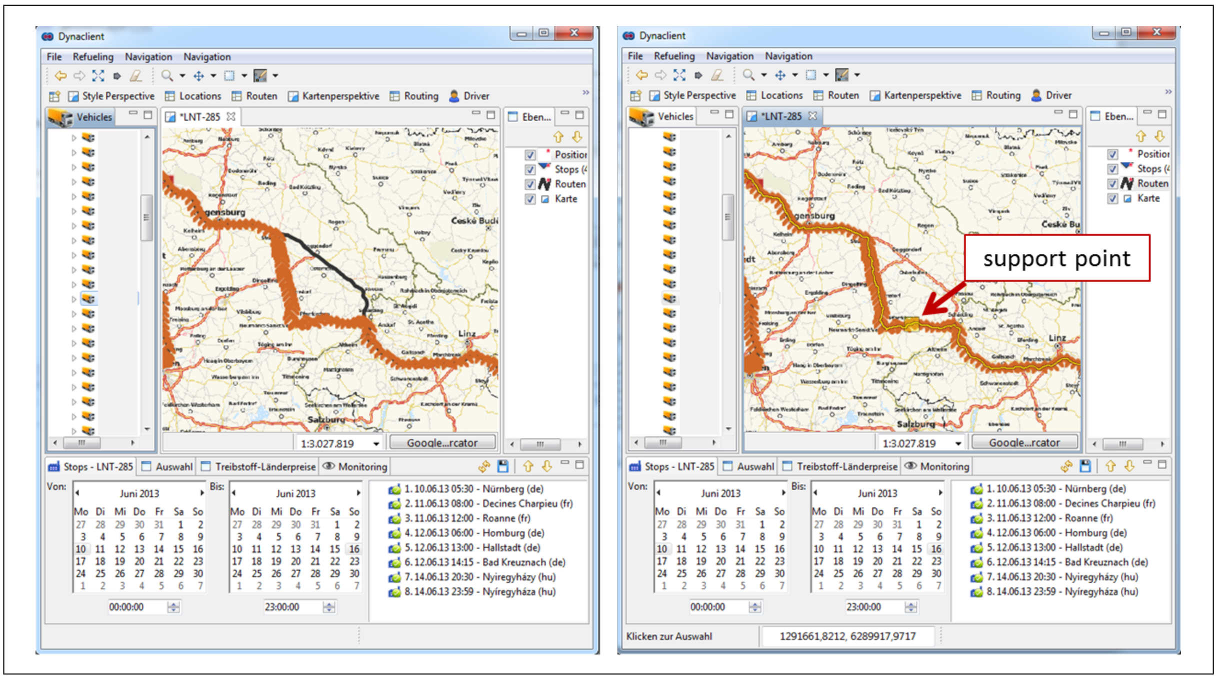

Figure 11: Fitting the computed route to the driver's route

A planning horizon of one week was considered. In all test cases, drivers start their tours on Monday after a weekly rest period. Varying starting times were taken into account. To be able to analyze the influence of the number of time windows on the run time, cases with one, two or three alternative time windows were generated. Exact time windows at customer locations were not available as they were chosen in customer systems and not transferred to the database of the prototype. Instead, planned arrival times extracted from the transport management software were used as a basis for the test instances. From process analysis it is known that arrival times were planned to be sooner rather than late. Therefore, for the cases with one or two time windows, the start of the first time window at a customer location was set to be equal to the planned arrival time. For the case with three alternative time windows, the start of the intermediate time window was set to be equal to the planned arrival time. Time window lengths of 0, 30, 60, 120 and 600 minutes were analyzed. The time between the end of a time window and the start of the subsequent time window was set to 120 minutes except for the instances with 600 minutes time windows. Here, a time interval of 840 minutes was chosen. These instances represent daily opening hours. Table 2 gives an overview of the underlying vehicle routes of the base instances.

[1] Support points, similar to the customer locations, have to be traversed by the route computed by the routing algorithm. They were used as a tool to fit the computed routes to the drivers' routes for which position data was available.

The values indicated in the last two columns correspond to the real distances traveled and the total real driving time.

base instance/ route	# stops including start and end	countries stops	total distance in km	overall driving duration in h
1	4	HU-DE-DE-ES	2914	36.85
2	5	ES-ES-DE-DE-IT	3391	42.48
3	6	DE-DE-DE-ES-ES-DE	3653	46.95
4	6	HU-DE-DE-IT-IT-DE	2831	36.45
5	6	FR-FR-DE-DE-HU-HU	1739	22.37
6	6	DE-DE-HU-DE-DE-IT	2944	37.47
7	7	ES-FR-FR-DE-DE-DE-DE	2269	30.32
8	7	DE-DE-IT-IT-DE-HU-HU	3142	39.77
9	7	IT-IT-IT-DE-DE-HU-IT	3019	38.17
10	8	DK-DK-DE-DE-DE-FR-FR-DK	3436	43.77
11	8	DE-IT-IT-DE-DE-FR-FR-DE	3447	43.62
12	9	HU-DE-DE-DE-FR-FR-FR-FR-DE	2475	31.85
13	10	FR-FR-FR-FR-FR-DE-DE-IT-IT-DE	2826	36.42
14	11	DE-DE-DE-DE-DE-FR-FR-DE-DE-DE-DE	3055	40.85
15	12	DE-DE-DE-DE-FR-FR-DE-DE-DE-DE-DE-DE	3250	41.95

Table 2: Characteristics of the test instances

For each base instance, 15 test instances were obtained by varying the number and length of time windows as described above. In total, 225 test instances were generated. Table 3 shows the number of variables and constraints for the test instances with three alternative time windows depending on the number of stops.

# stops (incl. start and end)	# binary variables	# integer (non-binary variables)	# continuous variables	# constraints
4	111	12	42	539
5	142	16	53	686
6	173	20	64	833
7	204	24	75	980
8	235	28	86	1127
9	266	32	97	1274
10	297	36	108	1421
11	328	40	119	1568
12	359	44	130	1715

Table 3: Total number of variables and constraints in the MILP model with optional rules (3 alternative time windows)

7.2 The solution of the model without optional rules as upper cutoff

In order to reduce the Cplex run time, the possibility to set an upper cutoff for the optimal solution value was examined. run times for the model without optional rules were known to be significantly shorter (see Section 7.3.3). Therefore, we used the optimal solution value obtained by solving the first submodel with a solution space restricted to possibilities that do not include optional rules as an upper cutoff. Figure 12 gives an overview of the solution processes with and without upper cutoff. Tables 4 and 5 display the run times for all test instances cumulated over 2 or 3 runs, respectively.

All instances were solved to optimality for both solution processes. On average, the run time was almost 39 % less when an upper cutoff was used. The two instances derived from base instance 15 with a time window length of 600 minutes and one or two time windows show the greatest improvements and are crucial for the reduced average run time. This can be identified clearly in Figure 13.

For further analysis of the run time, the solution process with three runs, that means with one additional run to determine an upper cutoff, was chosen.

7.3 Analysis

Figure 14 depicts the average run times of runs 1 to 3 depending on the number of customer vertices. As expected, the run time increases with the number of customer vertices. It is not surprising that the run time of the second run grows strongest, thus increasing the proportion of the second run on the overall run time from almost 42 % (4 stops) to more than 71 % (12 stops).

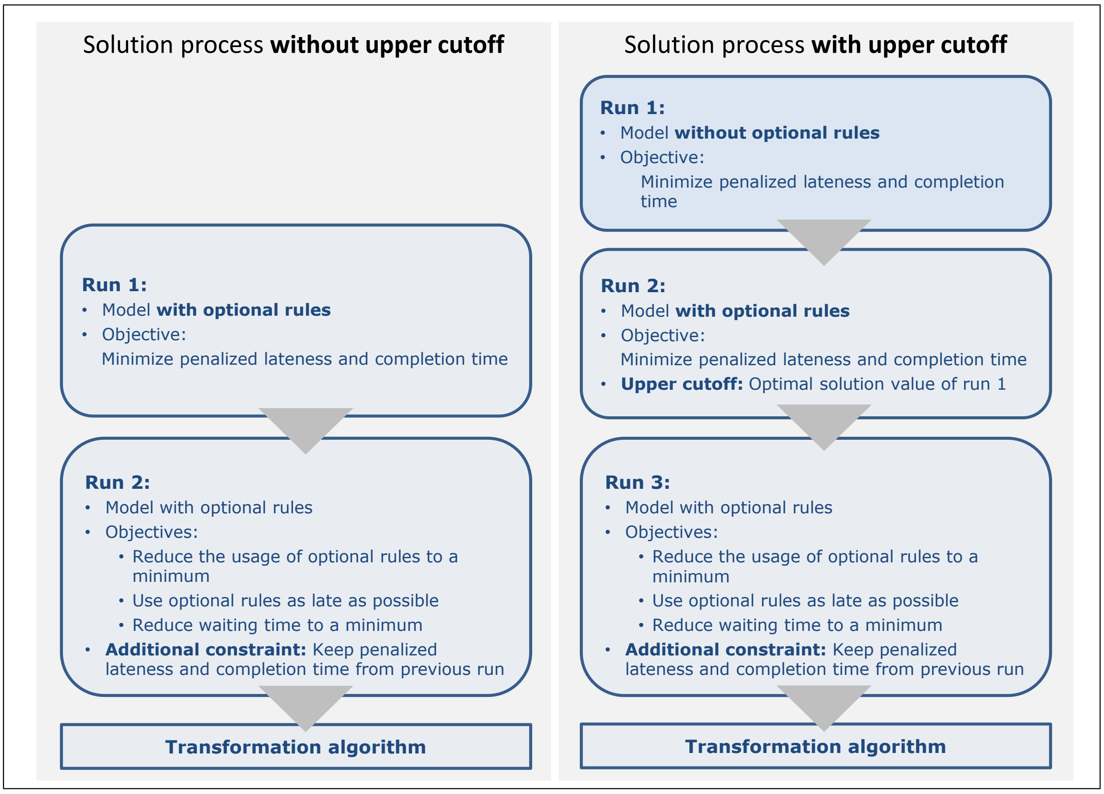

Figure 12: Solution processes with and without an upper cutoff

				length of time windows in min					
		base inst./ route	**# stops**	**0**	**30**	**60**	**120**	**600**	**Σ**
# time windows	**1**	**1**	4	0.094	0.125	0.078	0.032	0.048	0.377
		2	5	0.172	0.187	0.187	0.141	0.078	0.765
		3	6	0.062	0.327	0.077	0.078	0.157	0.701
		4	6	0.140	0.156	0.171	0.157	0.172	0.796
		5	6	0.079	0.094	0.078	0.078	0.155	0.484
		6	6	0.343	0.234	0.141	0.218	0.282	1.218
		7	7	0.265	0.203	0.265	0.313	0.219	1.265
		8	7	0.140	0.219	0.141	0.093	0.219	0.812
		9	7	0.109	0.125	0.125	0.171	0.219	0.749
		10	8	0.281	0.156	0.250	0.140	0.203	1.030
		11	8	0.702	0.468	0.328	0.467	0.593	2.558
		12	9	0.312	0.437	0.250	0.344	1.403	2.746
		13	10	2.512	0.717	0.265	0.342	2.901	6.737
		14	11	1.623	2.184	1.638	1.950	1.825	9.220
		15	12	1.233	3.806	3.011	1.778	79.748	89.576
	Σ			8.067	9.438	7.005	6.302	88.222	119.034
	2	**1**	4	0.031	0.032	0.030	0.047	0.031	0.171
		2	5	0.155	0.077	0.079	0.063	0.063	0.437
		3	6	0.062	0.077	0.063	0.079	0.093	0.374
		4	6	0.187	0.140	0.140	0.202	0.266	0.935
		5	6	0.109	0.109	0.187	0.202	0.187	0.794
		6	6	0.124	0.249	0.296	0.234	0.266	1.169
		7	7	0.234	0.344	0.234	0.187	0.405	1.404
		8	7	0.140	0.155	0.235	0.171	0.265	0.966
		9	7	0.172	0.187	0.156	0.281	0.109	0.905
		10	8	0.561	0.420	0.328	0.422	0.359	2.090
		11	8	0.717	0.687	0.671	0.640	0.171	2.886
		12	9	0.655	1.014	0.359	0.765	0.811	3.604
		13	10	0.266	0.343	0.531	0.546	2.496	4.182
		14	11	2.278	1.794	2.277	0.765	1.809	8.923
		15	12	0.796	2.043	1.778	7.223	44.461	56.301
	Σ			6.487	7.671	7.364	11.827	51.792	85.141
	3	**1**	4	0.062	0.062	0.046	0.077	0.047	0.294
		2	5	0.078	0.063	0.078	0.078	0.188	0.485
		3	6	0.078	0.079	0.094	0.094	0.421	0.766
		4	6	0.328	0.265	0.186	0.125	0.265	1.169
		5	6	0.187	0.155	0.171	0.173	0.110	0.796
		6	6	0.358	0.546	0.609	0.282	0.437	2.232
		7	7	0.217	0.202	0.313	0.748	0.516	1.996
		8	7	0.250	0.359	0.125	0.390	0.375	1.499
		9	7	0.203	0.156	0.172	0.172	0.577	1.280
		10	8	0.530	1.170	0.546	0.234	1.170	3.650
		11	8	2.043	1.123	0.780	0.795	1.903	6.644
		12	9	0.546	1.232	2.013	1.030	1.373	6.194
		13	10	1.092	0.593	2.340	1.248	2.340	7.613
		14	11	3.479	3.962	2.168	3.993	10.514	24.116
		15	12	2.153	3.246	4.228	2.558	9.032	21.217
	Σ			11.604	13.213	13.869	11.997	29.268	79.951

Table 4: Run times in seconds for the MILP model without additional run for an upper cutoff

				length of time windows in min					
		base inst./ route	**# stops**	**0**	**30**	**60**	**120**	**600**	**Σ**
		1	4	0.064	0.141	0.047	0.063	0.078	0.393
		2	5	0.109	0.203	0.218	0.109	0.109	0.748
		3	6	0.063	0.062	0.077	0.079	0.125	0.406
		4	6	0.110	0.172	0.124	0.094	0.155	0.655
		5	6	0.126	0.095	0.095	0.095	0.125	0.536
		6	6	0.126	0.140	0.125	0.140	0.188	0.719
		7	7	0.311	0.295	0.375	0.281	0.265	1.527
	1	**8**	7	0.125	0.123	0.141	0.140	0.312	0.841
		9	7	0.125	0.125	0.188	0.126	0.110	0.674
		10	8	0.764	0.280	0.405	0.437	0.328	2.214
		11	8	0.436	0.687	0.344	0.328	0.937	2.732
		12	9	0.592	0.934	0.531	0.545	0.907	3.509
		13	10	0.809	1.077	0.904	0.609	1.575	4.974
		14	11	1.997	2.168	1.826	1.685	1.576	9.252
		15	12	1.529	1.637	9.874	2.059	3.962	19.061
	Σ			7.286	8.139	15.274	6.790	10.752	48.241
		1	4	0.077	0.062	0.062	0.079	0.063	0.343
		2	5	0.108	0.077	0.079	0.094	0.126	0.484
		3	6	0.078	0.062	0.062	0.095	0.157	0.454
		4	6	0.172	0.203	0.186	0.172	0.203	0.936
		5	6	0.109	0.717	0.094	0.141	0.203	1.264
		6	6	0.154	0.218	0.265	0.343	0.173	1.153
		7	7	0.266	0.311	0.250	0.203	0.219	1.249
# time windows	**2**	**8**	7	0.172	0.171	0.155	0.124	0.172	0.794
		9	7	0.156	0.125	0.156	0.234	0.218	0.889
		10	8	0.576	0.375	0.375	0.437	0.312	2.075
		11	8	0.843	0.764	0.437	0.547	0.374	2.965
		12	9	1.265	0.951	0.779	1.091	0.905	4.991
		13	10	0.499	0.514	0.968	0.749	1.810	4.540
		14	11	2.137	1.576	1.389	0.717	2.324	8.143
		15	12	3.058	2.029	2.168	8.767	8.221	24.243
	Σ			9.670	8.155	7.425	13.793	15.480	54.523
		1	4	0.077	0.109	0.078	0.095	0.030	0.389
		2	5	0.110	0.141	0.077	0.093	0.265	0.686
		3	6	0.078	0.063	0.094	0.110	0.359	0.704
		4	6	0.266	0.311	0.219	0.141	0.219	1.156
		5	6	0.375	0.156	0.126	0.142	0.110	0.909
		6	6	0.421	0.437	0.578	0.249	0.265	1.950
		7	7	0.344	0.234	0.281	0.373	0.406	1.638
	3	**8**	7	2.058	0.296	0.186	0.422	0.203	3.165
		9	7	0.156	0.140	0.188	0.204	0.390	1.078
		10	8	0.655	0.422	0.469	0.453	0.701	2.700
		11	8	1.996	1.435	1.030	0.703	1.686	6.850
		12	9	0.889	1.373	1.403	1.404	1.139	6.208
		13	10	1.014	1.201	1.420	1.139	3.416	8.190
		14	11	4.025	3.402	2.027	2.278	10.935	22.667
		15	12	2.028	1.358	2.465	3.509	6.973	16.333
	Σ			14.492	11.078	10.641	11.315	27.097	74.623

Table 5: Run times in seconds for the MILP model with additional run for an upper cutoff

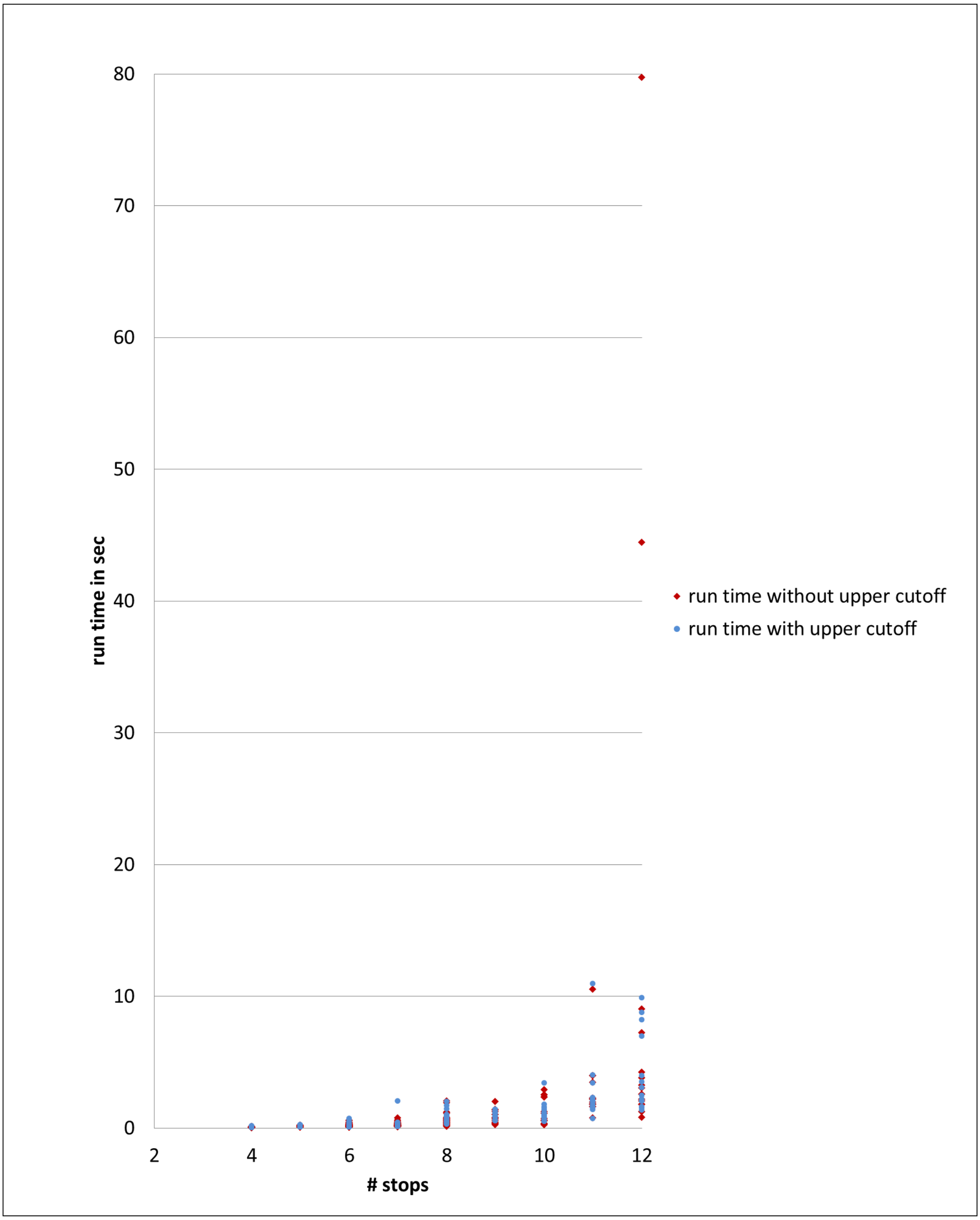

Figure 13: Run times for the MILP model with optional rules

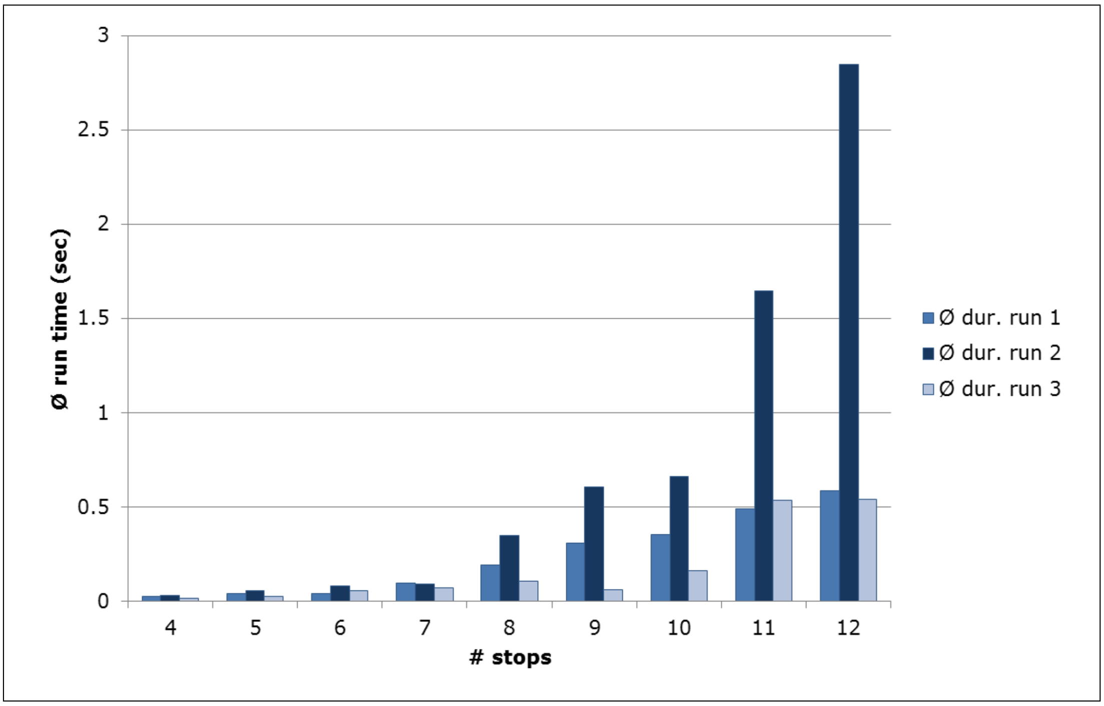

Figure 14: Average run time depending on the number of vertices

7.3.1 The influence of the number of time windows on the run time

Figure 15 shows the influence of the number of time windows on the average run time depending on the number of vertices. In the following examples it clearly can be seen that the number of time windows per stop as well as the number of vertices are important for the run time.

Example 1: 10 stops, 2 time windows:

- Total number of variables: 433.
- Number of constraints: 1421.
- Average run time: 0.908 s

Example 2: 11 stops, 2 time windows:

- Total number of variables: 478. Increase compared to Example 1: 10 %.
- Number of constraints: 1568. Increase compared to Example 1: 10 %.
- Average run time: 1.629 s. Increase compared to Example 1: 79 %.

Example 3: 10 stops, 3 time windows:

- Total number of variables: 441. Increase compared to Example 1: 2 %.

- Number of constraints: 1421. Increase compared to Example 1: 0 %.
- Average run time: 1.638 s. Increase compared to Example 1: 80 %.

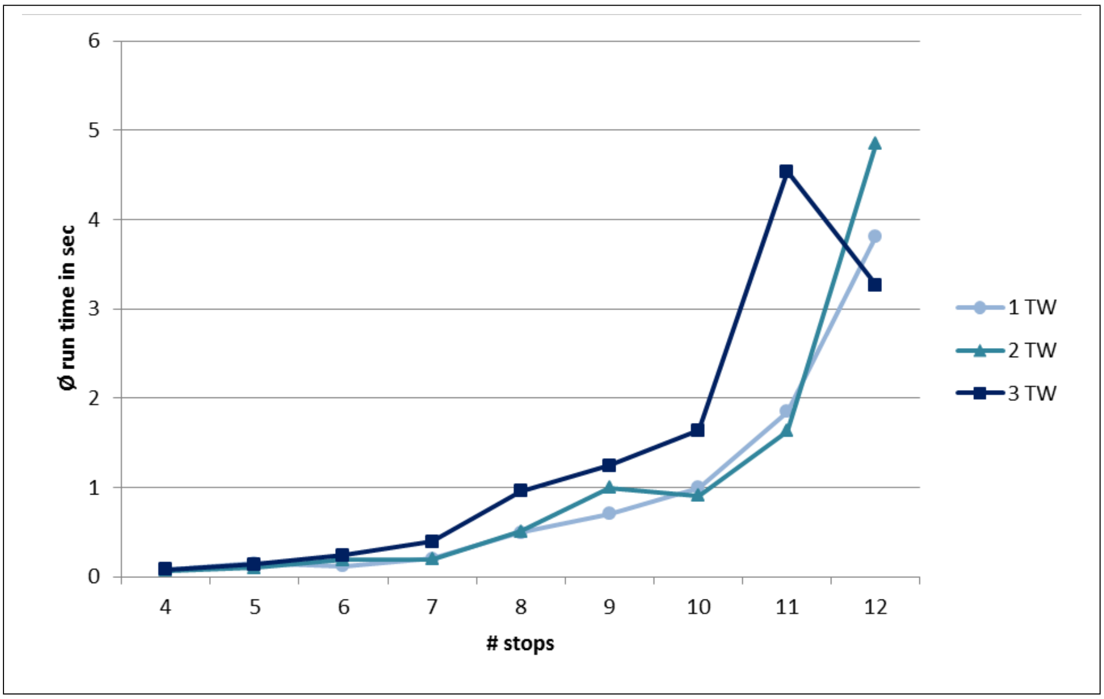

Figure 15: The influence of the number of time windows (TW) on the run time

7.3.2 The influence of the time window length on the run time

The three diagrams in Figure 16 show the average run times depending on the number of stops and the time window length separately for each number of time windows.

Figure 17 depicts the average run times of all numbers of alternative time windows.

We can see from Figure 16 that there are significantly long run times in some of the instances with a time window length of 10 h compared to the other instances. Furthermore, there is one test instance (there is only one base instance that has 12 stops) with a time window length of 120 minutes that has a comparably long run time and another one with a time window length of 60 minutes. Test instances with time window lengths of 30 minutes or exact arrival times performed best. If we do consider the cumulated view (Figure 17), in some cases, we can see significant decreasing average run times if a time window with a length of 30 minutes instead of an exact arrival time is possible. In others, a slight increase can be recognized.

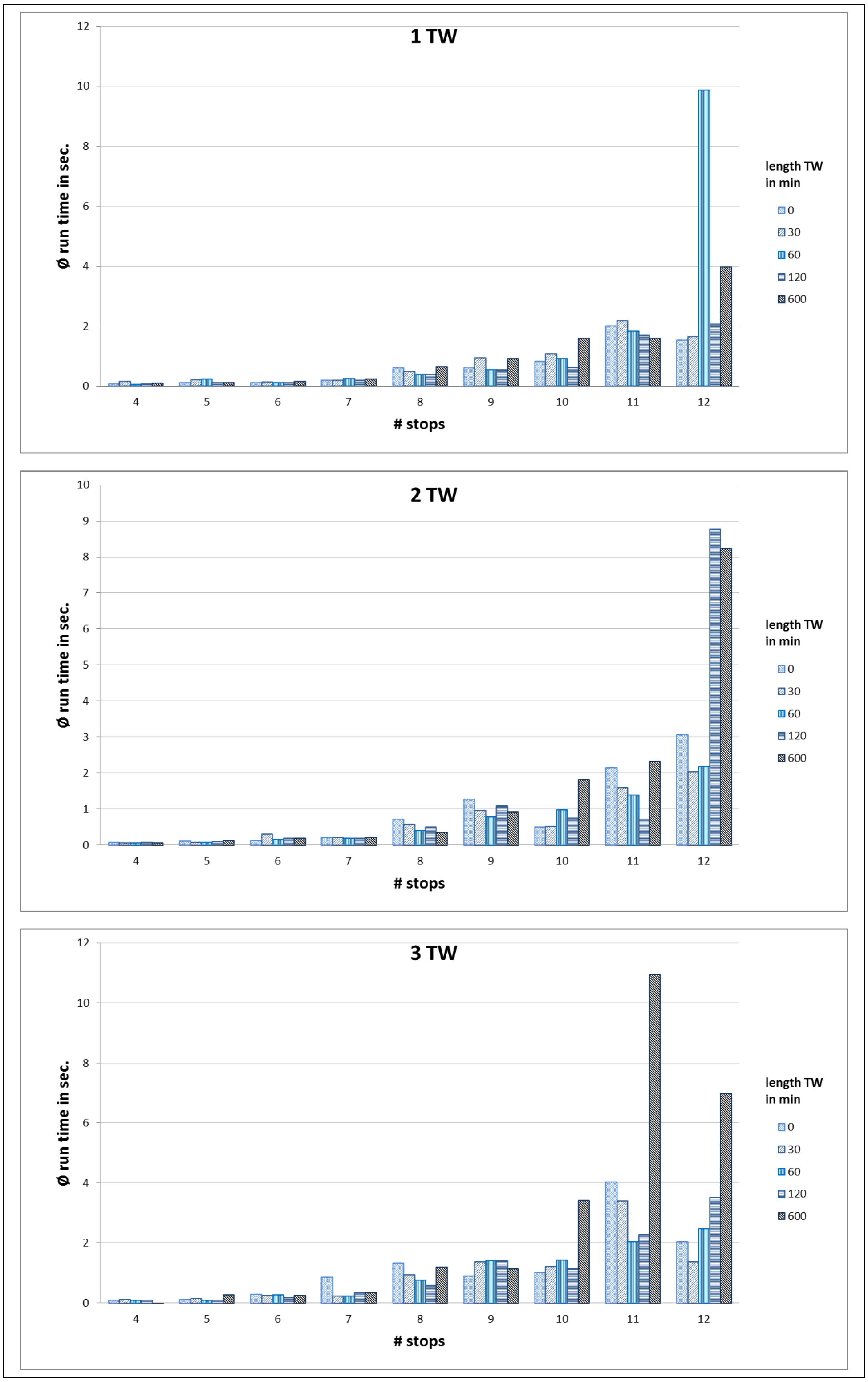

Figure 16: The influence of the time window length on the run time

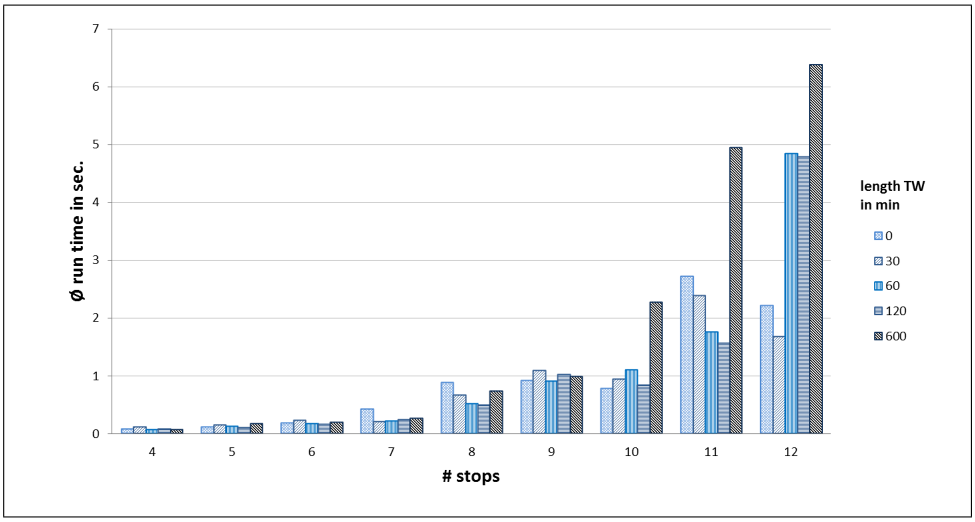

Figure 17: The influence of the time window length on the run time

7.3.3 The influence of the optional rules

In the following, we examine the influence of the optional rules on the run time and the reachability of time windows. For the case without optional rules, constraints (6.187) - (6.202) are included. For the case with optional rules, the solution process with upper cutoff is chosen (see Figure 12). Table 6 shows the cumulated run time of both runs for each of the 225 test instances for the model without optional rules. Figure 18 compares the average run time for each base instance with the run time of the MILP model with optional rules (see also Figure 5 on page 76).

Ignoring the optional rules, the average run time of the test instances per base instance could be reduced by 37 % (base instance 1, 4 stops) up to 81 % (base instance 15, 12 stops). This means that the optional rules have a strong influence on the overall run time.

Lateness, on the other hand, is reduced significantly if optional rules are considered. Figure 19 shows the average lateness per base instance. The large difference for base instance 12 is notable. Here, time windows very often cannot be met at all and in some instances lateness at a single customer location of sometimes more than 10 hours cannot be avoided. In practice, re-planning would be necessary. Only in the instances with simulated opening hours and at least two time windows lateness can be avoided, no matter if the optional rules are used or not. When considering the optional rules, the largest share of the lateness reduction is achieved because in some of the instances a complete daily rest period can be left out. In these instances, this has an effect on the lateness at several customer locations. For base instances 1 and 2 no lateness occurred even if optional rules were ignored. For base instances 4 and 9, the average lateness was not improved when considering the optional rules. In all other cases, a reduction was noticed. Cumulating the lateness over all test instances, a reduction of 55 % can be observed by taking into account optional rules.

				length of time windows in min					
		base inst./ route	**# stops**	**0**	**30**	**60**	**120**	**600**	**Σ**
# time windows	**1**	**1**	4	0.032	0.046	0.046	0.047	0.032	0.203
		2	5	0.031	0.031	0.046	0.046	0.077	0.231
		3	6	0.048	0.032	0.046	0.031	0.046	0.203
		4	6	0.047	0.062	0.032	0.079	0.063	0.283
		5	6	0.063	0.031	0.046	0.031	0.077	0.248
		6	6	0.094	0.031	0.047	0.063	0.047	0.282
		7	7	0.094	0.077	0.094	0.094	0.061	0.420
		8	7	0.062	0.062	0.063	0.062	0.094	0.343
		9	7	0.062	0.032	0.079	0.030	0.078	0.281
		10	8	0.078	0.110	0.124	0.141	0.125	0.578
		11	8	0.125	0.282	0.234	0.219	0.281	1.141
		12	9	0.484	0.390	0.608	0.468	0.406	2.356
		13	10	0.266	0.296	0.624	0.343	0.250	1.779
		14	11	0.405	0.437	0.327	0.235	0.872	2.276
		15	12	0.469	0.795	0.935	0.531	0.249	2.979
	Σ			2.360	2.714	3.351	2.420	2.758	13.603
	2	**1**	4	0.063	0.046	0.032	0.046	0.031	0.218
		2	5	0.031	0.047	0.062	0.078	0.062	0.280
		3	6	0.047	0.046	0.032	0.046	0.063	0.234
		4	6	0.063	0.047	0.093	0.079	0.093	0.375
		5	6	0.047	0.078	0.047	0.062	0.093	0.327
		6	6	0.079	0.062	0.078	0.109	0.063	0.391
		7	7	0.093	0.110	0.125	0.063	0.078	0.469
		8	7	0.078	0.047	0.047	0.047	0.110	0.329
		9	7	0.078	0.062	0.063	0.156	0.109	0.468
		10	8	0.140	0.218	0.203	0.125	0.187	0.873
		11	8	0.297	0.281	0.624	0.203	0.281	1.686
		12	9	0.577	0.515	0.421	0.593	0.374	2.480
		13	10	0.297	0.296	0.343	0.187	0.343	1.466
		14	11	0.234	0.312	0.373	0.344	0.483	1.746
		15	12	1.029	1.061	0.748	0.780	0.375	3.993
	Σ			3.153	3.228	3.291	2.918	2.745	15.335
	3	**1**	4	0.063	0.094	0.047	0.031	0.047	0.282
		2	5	0.063	0.063	0.031	0.079	0.063	0.299
		3	6	0.046	0.031	0.016	0.079	0.124	0.296
		4	6	0.079	0.079	0.110	0.094	0.094	0.456
		5	6	0.030	0.062	0.046	0.062	0.046	0.246
		6	6	0.077	0.141	0.078	0.109	0.110	0.515
		7	7	0.157	0.110	0.078	0.062	0.063	0.470
		8	7	1.982	0.171	0.110	0.109	0.391	2.763
		9	7	0.093	0.093	0.094	0.094	0.061	0.435
		10	8	0.219	0.172	0.218	0.312	0.312	1.233
		11	8	0.639	0.515	0.156	0.531	0.312	2.153
		12	9	0.655	0.625	0.842	0.764	0.515	3.401
		13	10	0.515	0.436	0.983	0.297	0.811	3.042
		14	11	0.780	0.780	0.951	0.530	1.154	4.195
		15	12	0.638	0.484	0.889	0.638	1.264	3.913
	Σ			6.036	3.856	4.649	3.791	5.367	23.699

Table 6: Run times in seconds for the MILP model without optional rules

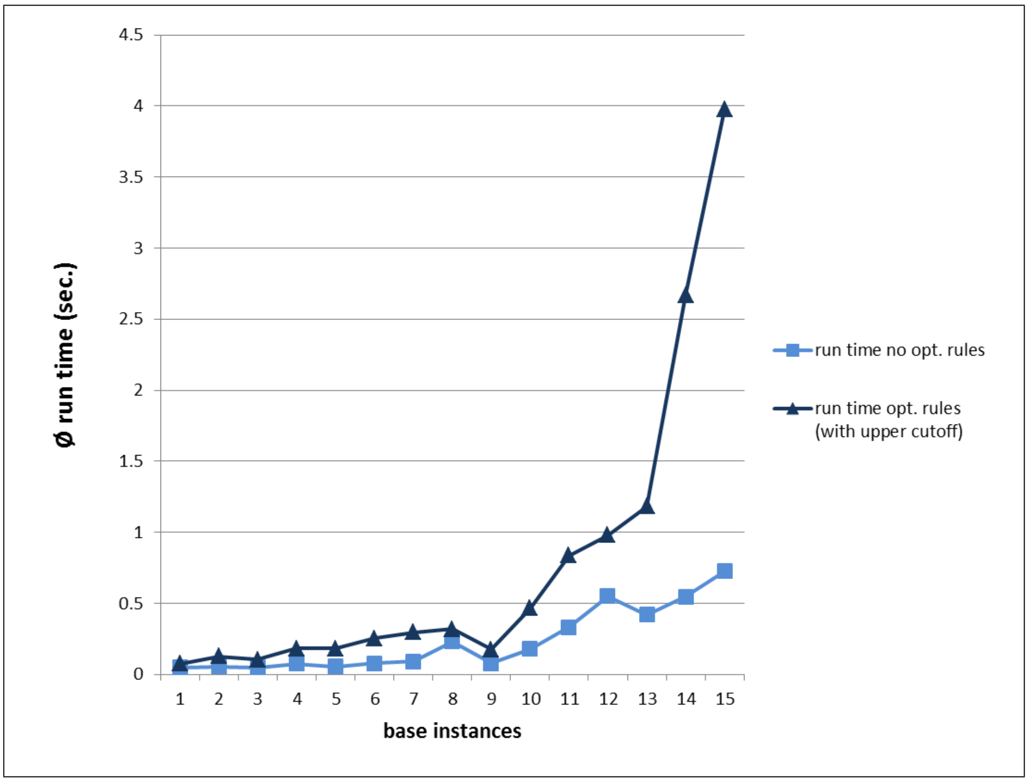

Figure 18: Average run times with and without considering optional rules

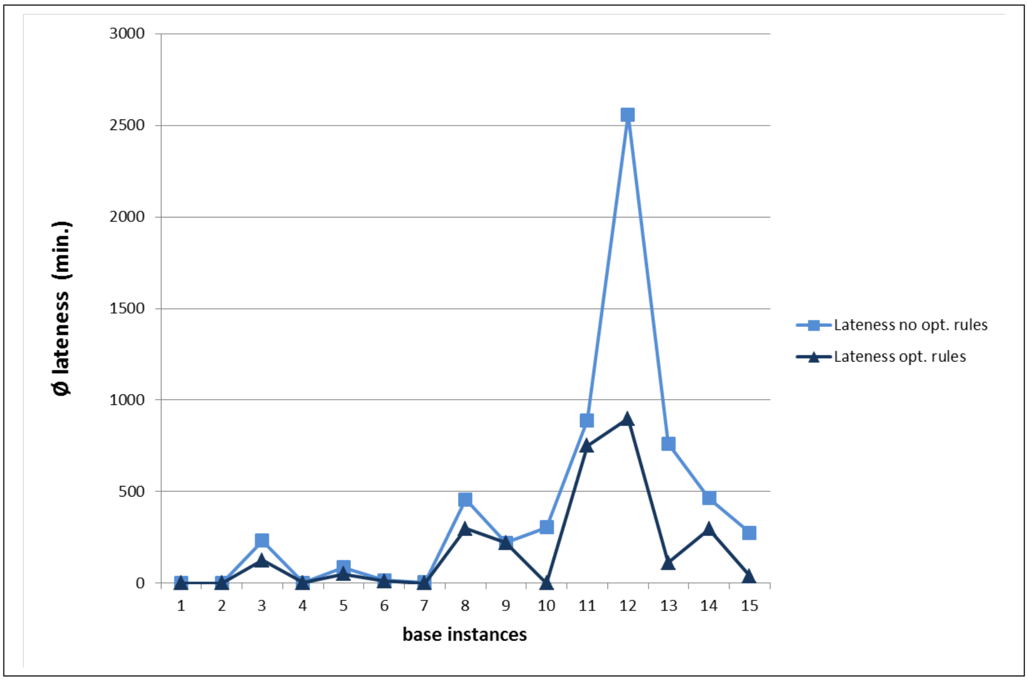

Figure 19: Average lateness with and without considering optional rules

The influence of the optional rules on the overall *travel time*, that means, the overall time from the start of the schedule until the last stop is reached, should not be neglected as well. Figure 20 shows the average travel time with and without optional rules per base instance. The computational results show that on average over all test instances the travel time is reduced by 5 % when optional rules are considered. The average reduction of the travel time is nearly 6 hours (357 minutes), this means that the last stop can be reached much earlier. Thus, the start of the weekly rest period can be earlier if optional rules are taken into account. This option is interesting if drivers return home for the weekend. If not, as it is the case in the test instances, they may continue driving and may reach the next customer earlier. Furthermore, drivers may use a part of the additional time to look for a good rest area to stay for the weekend.

Summing up, it is really worthwhile to consider the optional rules when planning driver schedules. The test scenarios presented so far do not consider unexpected events like for example traffic jams, but such events may extend the travel times significantly. Therefore, it is important to include time buffers, for example, in the driving durations between customers and in the loading and unloading times, no matter which planning technique is chosen. This can be done by multiplying the durations with a constant factor that is greater than one and using the result as new estimated duration. Thus, time buffers are obtained that are proportional to the durations. Other techniques that, for example, incorporate road data and the probability of traffic jams for different roads are possible.

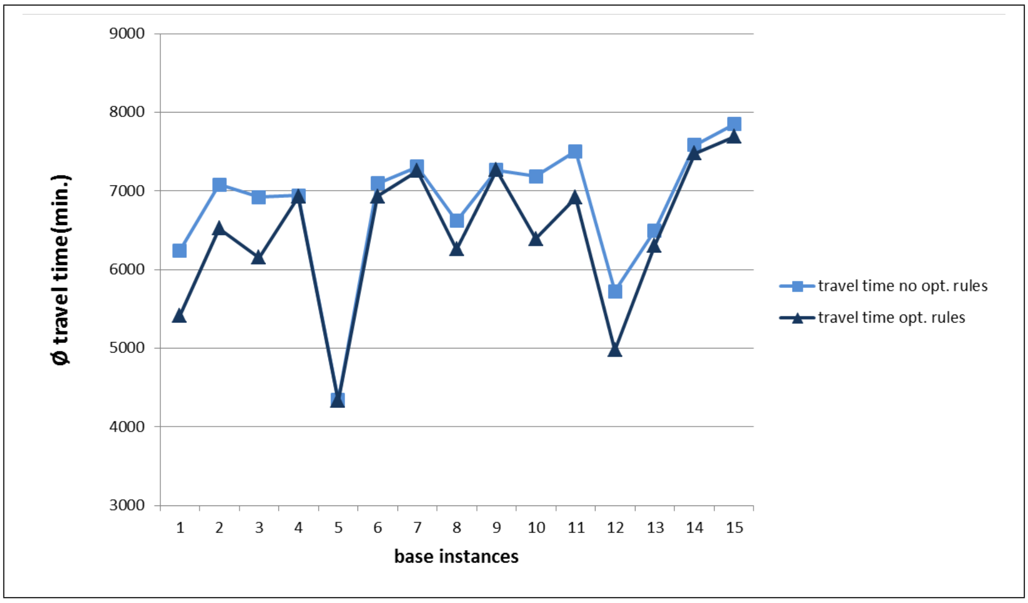

Figure 20: Average travel time with and without considering optional rules

8 Myopic algorithm - A heuristic

The MILP models presented in the previous sections allow us to set up an optimal driver schedule for driver rest periods and breaks and to choose time windows. To solve the models efficiently, advanced optimization algorithms are necessary and an optimization solver is recommended to obtain high-quality solutions within a reasonable time. Optimization software has to be integrated and acquisition costs must not be neglected. A transport company has to weigh the additional costs against the advantages and cost savings achievable.

To get a feeling for the magnitude of the schedule improvement, a myopic algorithm was developed that mimes a dispatcher that uses sophisticated strategies to plan driver activities considering the routes between customer locations and time windows successively one after another.

The input for the algorithm is the same as for the models presented: the driver status at the beginning of the planning horizon, the sequence of customer locations and other stops to be visited and driving durations between consecutive stops i and $i+1$ ($\bar{\Delta}^{drive}_{(i,i+1)}$), start and end times of possible time windows at stop i ($\overline{TW}^{begin}_{i,z}$, $\overline{TW}^{end}_{i,z}$) and planned durations for handling activities including loading and/or unloading ($\bar{\Delta}^{service}_{i}$). Durations are expressed in minutes. Time windows are given in minutes since the end of the last weekly rest period.

Similar to the naive labeling algorithm proposed by Goel (2009), the driver status is represented by an n-tupel. The three-tupel described in Goel (2009) that defines the driver status at arrival at a stop contains the arrival time, the cumulated driving time since the last (daily or weekly) rest period and the cumulated driving time since the last break or rest period (nonstop driving time). The myopic algorithm, similar to the naive labeling algorithm, only considers at each decision point the driver status and activities concerning the current arc (i.e. decisions about activities that take place between leaving stop i and loading or unloading at stop $i+1$) and decides for exactly one alternative. Once the plan for an arc is made, the driver status at the subsequent stop is fixed and the algorithm proceeds with planning the next arc.

8.1 The driver status

We extend the tupel representing the driver status to be able to additionally consider the optional rules. To this end, the following parameters are considered:

- *ptwr*:
  Time elapsed since the last weekly rest period in minutes.
- *udt*:
  Uninterrupted driving time since the last break or daily rest period.
- *ddt*:
  Cumulated driving time since the last daily or weekly rest period.
- *ptr*:
  Time elapsed since the end of the last daily or weekly rest period.
- *hpb*:
  Takes the value 1 if a partial break was taken and 0 otherwise.
- *hpr*:
  1 if a partial daily rest period has been taken, 0 otherwise
- *noRed*:
  Number of reduced daily rest periods taken since the end of the last weekly rest period. (Previous daily rest periods with a duration of less than 11 hours).
- *noExt*:
  Number of extended daily driving times already taken in this week. If the current daily driving time exceeds 9 hours, this information is included.
- *red*:
  1 if a driving time extension is active (more than 9 hours of daily driving time), 0 otherwise.
- *dte*:
  1 if the next daily rest period is planned to be a reduced one, 0 otherwise.

The driver status at the beginning of the planning horizon is given by the tupel

$$driverStatus = (\overline{ptwr}, \overline{udt}, \overline{ddt}, \overline{ptr}, \overline{hpb}, \overline{hpr}, \overline{noRed}, \overline{noExt}, \overline{red}, \overline{dte}).$$

The corresponding status variables are initialized accordingly (see Section B Algorithm 2 in the appendices).

8.2 Updating the driver status

Each activity has an activity type and a duration and when added to the schedule it modifies the driver status. Activity types considered by the myopic algorithm are:

- *rest*: Regular, first or second part of a daily rest period
- *redrest*: Reduced daily rest period
- *drive*: Driving
- *work*: Loading or unloading
- *break*: Break
- *wait*: Wait

More details about the update algorithm for the driver status can be found in the appendices (see Section B Algorithm 3).

8.3 The algorithm

The myopic algorithm is structured as follows. At first, the driver status is initialized. Then, activities between each pair of successive stops i and $i+1$ are scheduled sequentially. This is done in three steps. In step one, activities between stops i and $i+1$ are scheduled. Step two is concerned with the choice of the time window at the next stop $i+1$. In step three, activities at stop $i+1$ are scheduled. Steps one to three are repeated until the last stop is reached. Figure 21 shows a flowchart of the complete algorithm. In the following, more detailed flowcharts and short descriptions are given for the different steps. The corresponding pseudo-code can be found in the appendices (see Algorithms 5 - 7).

The procedure in the step **"Scheduling driver activities on arc (i,i+1)"** is similar to the naive method for scheduling driving periods, breaks, and rest periods presented by Goel (2009) and is illustrated in Figure 22. In addition to Goel (2009) we consider the possibility to extend the daily driving time by one hour.

In the step **"Choose time window at stop i+1"**, the time window at the next stop $i+1$ is chosen. Therefore, the first reachable time window, that means the first time window that ends after the last activity scheduled so far, is determined. When considering the current schedule, a daily rest period is necessary as the time left does not suffice to wait, load and/or unload at customer location $i+1$ because of the maximum time interval between two daily rest periods, it is tried to reduce the durations of daily rest periods scheduled on the arc $(i, i+1)$. If this does not help to be on time, it is tested if it is possible to extend the duration of the previous daily rest period and thus eliminate waiting time. In this way, the start of the time interval between the last daily rest period and the next one that is not scheduled yet can be postponed. The option to plan the next daily rest period to be a reduced one is taken into account.[1] If with this modification the daily rest period can take

[1] If the next daily rest period is planned to be a reduced one, the maximum time between the last daily rest period and the following one increases by 2 hours.

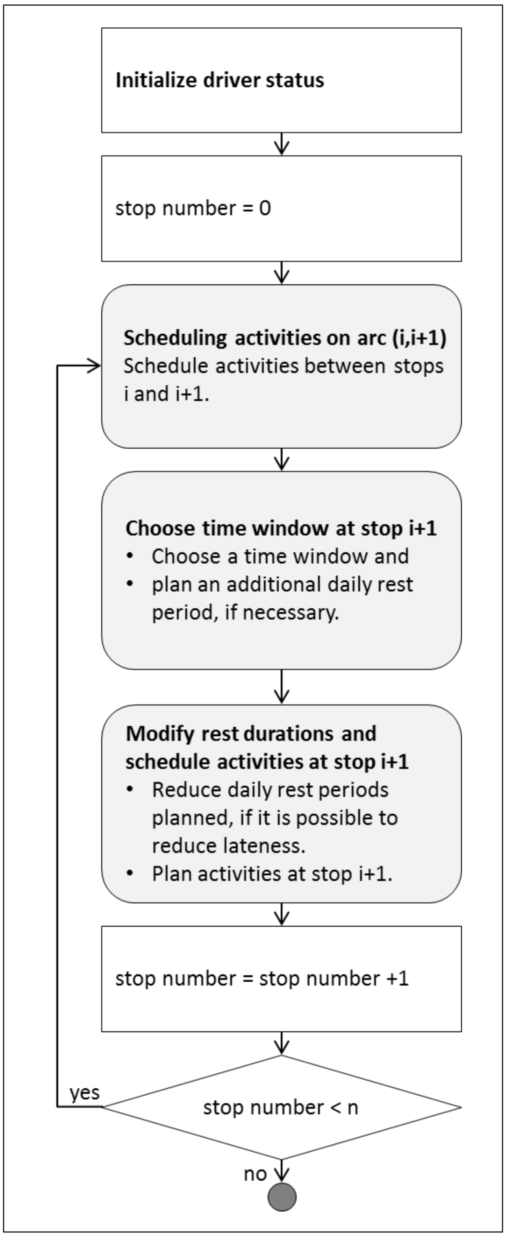

Figure 21: Myopic algorithm - flowchart

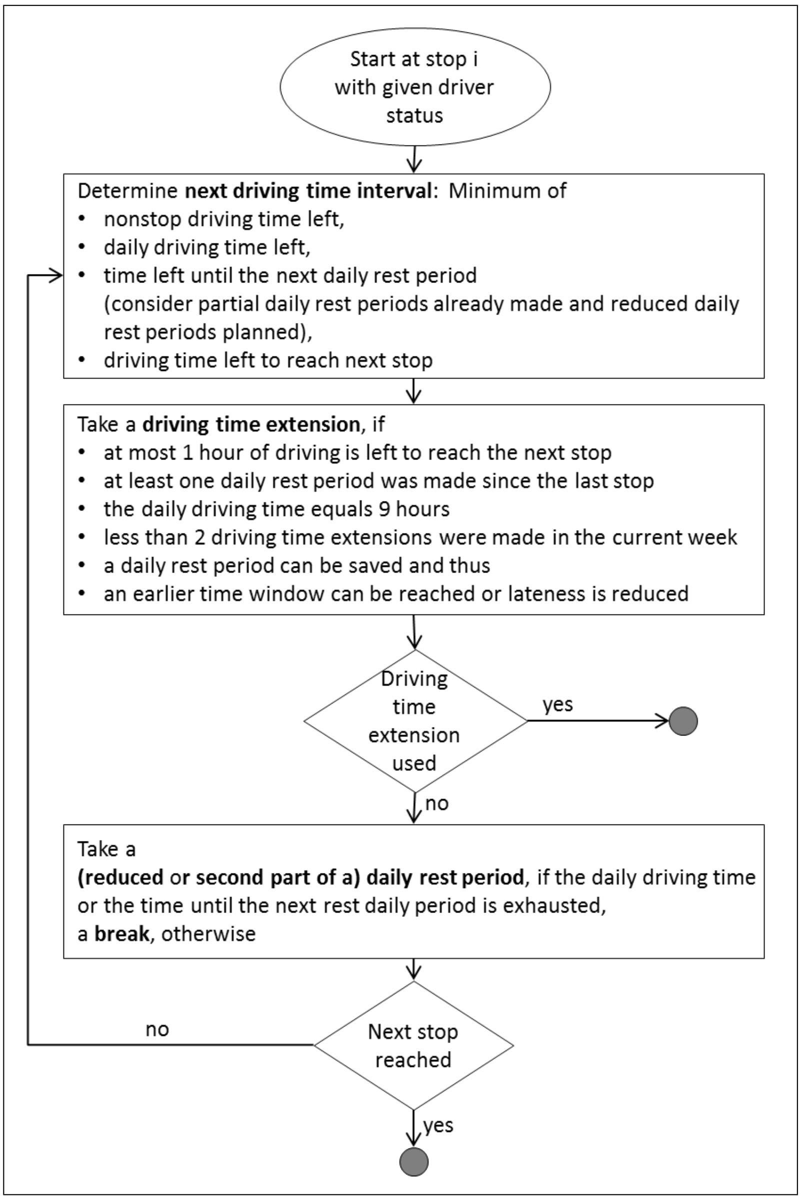

Figure 22: Scheduling activities on arc $(i, i+1)$

place after loading and/or unloading at the customer, the schedule is altered accordingly. In case the above modifications do not suffice to reach the time window, the next time window[1] is selected. These steps are repeated if either it is possible to be on time or the last time window is selected. The flowchart in Figure 23 illustrates the course of action.

In the step **"Modifying rest durations and scheduling activities at stop i+1"** (Figure 24), if there is lateness, regular daily rest periods are reduced if possible, and activities are rescheduled accordingly. Afterwards, activities at stop $i + 1$ are scheduled. If there is still time left until the start of the time window chosen, potential waiting time can be compensated by a resting activity. The options to take a partial daily rest period or a partial break are included. If there is still waiting time, we try to compensate it by extending the last daily rest period on the current arc. Finally, if the stop is a customer location, loading and/or unloading is scheduled.

[1] Time windows should be sorted by their start time and should not overlap.

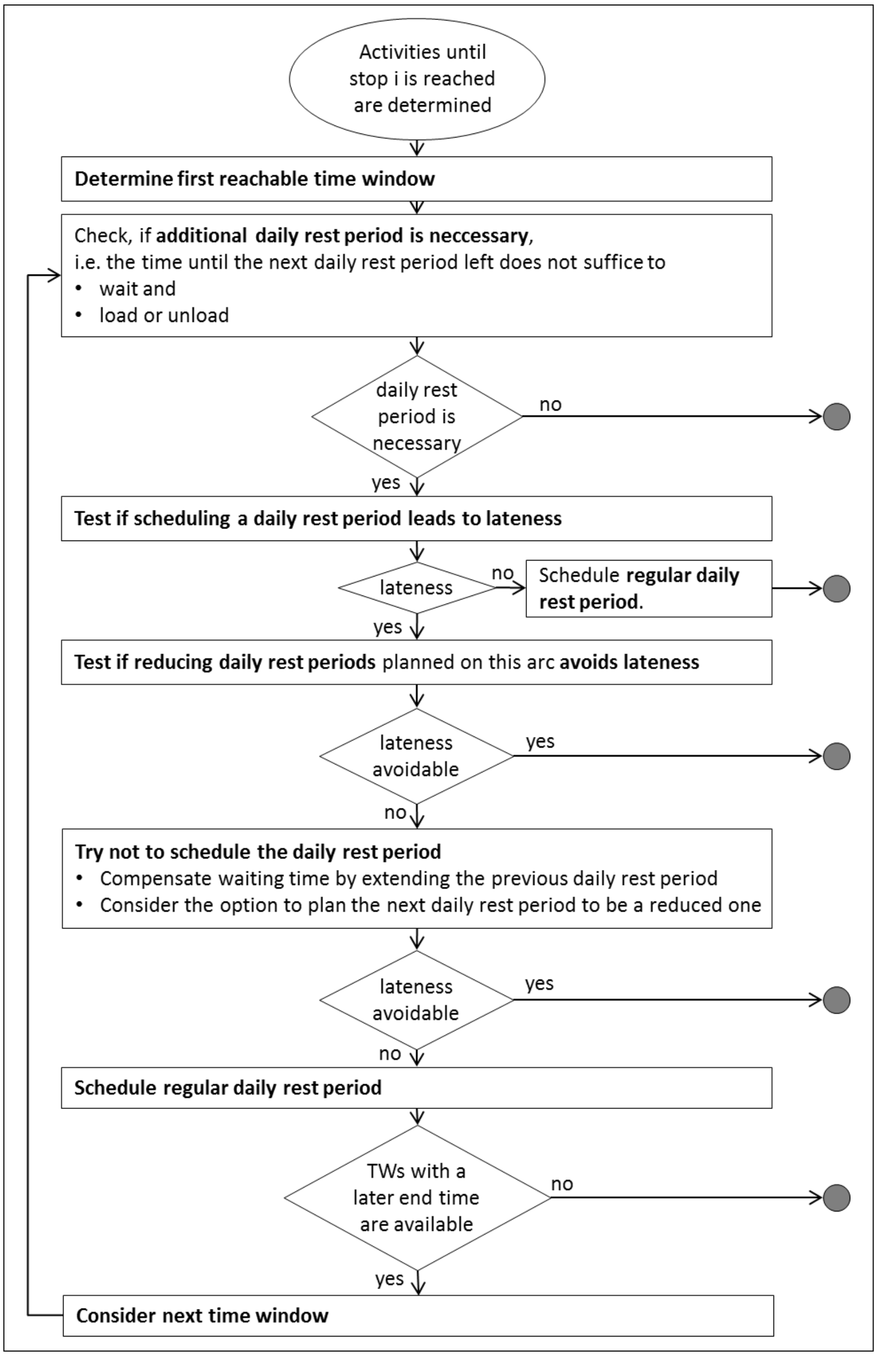

Figure 23: Choose time window at stop $(i, i + 1)$

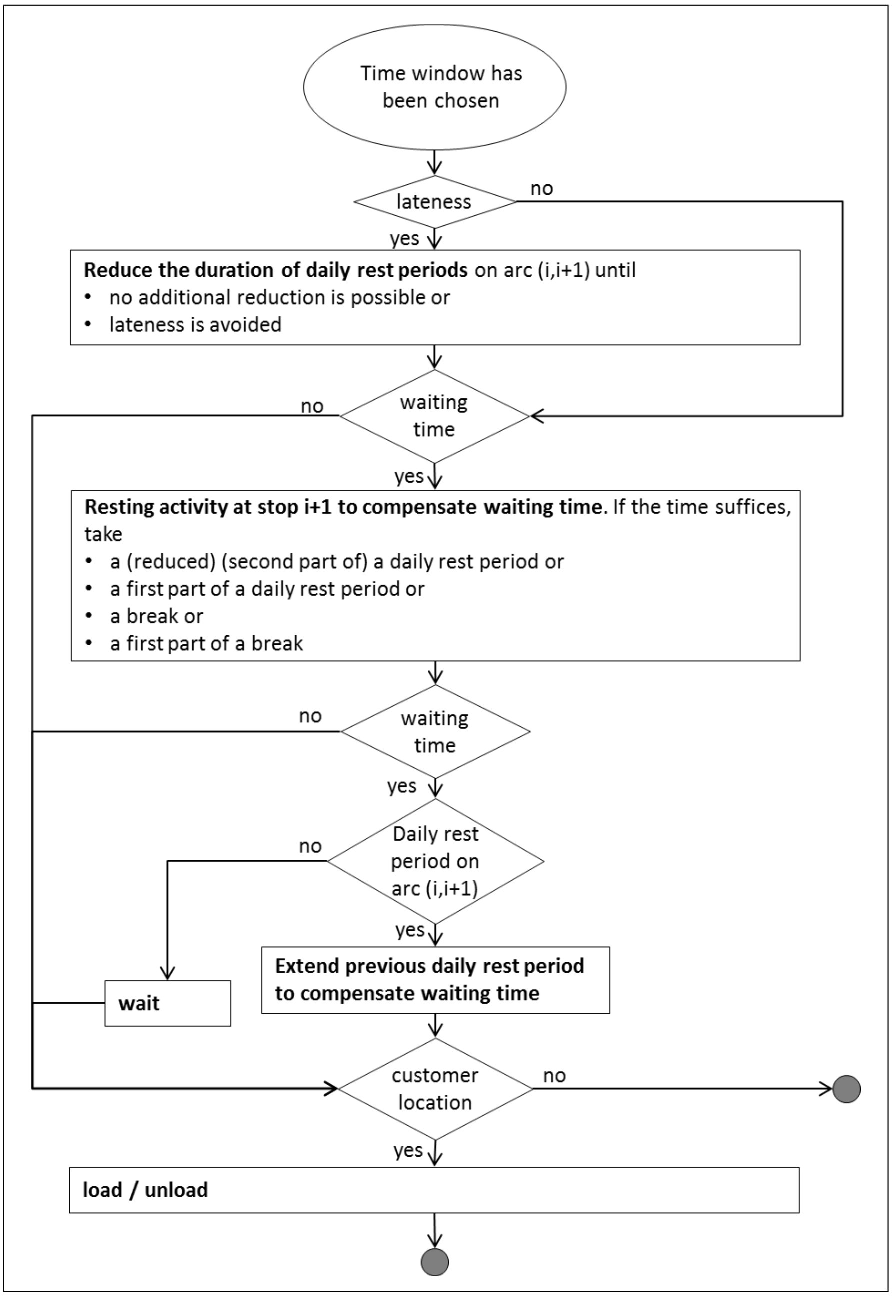

Figure 24: Modifying rest durations and scheduling activities at stop $i + 1$

9 Numerical experiments - Part 2

In this section we present the results obtained with the myopic algorithm and compare them to the results achieved with the MILP models. General findings are discussed at the end of the section.

9.1 Comparison of the algorithm and the MILP models

The run time of the myopic algorithm was less than 1 millisecond for each test instance. In Figures 25 and 26, lateness and overall travel times of the schedules constructed by the myopic algorithm and those constructed with the MILP models (with and without optional rules) are compared.

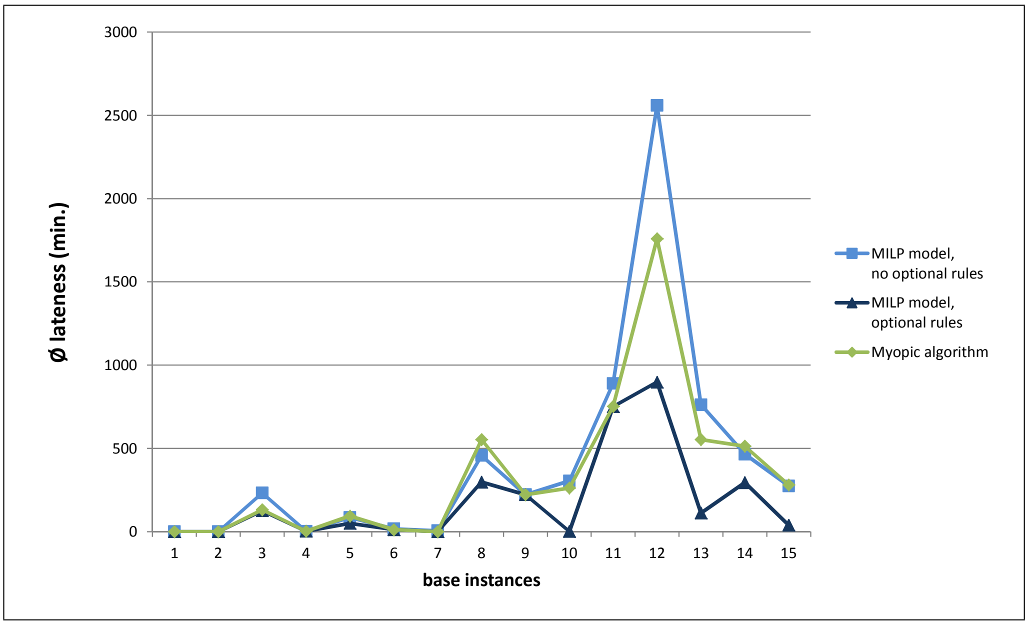

Figure 25: Average lateness of schedules depending on the solution technique

Figure 25 shows that, on average, the myopic algorithm performs better than the model without optional rules. Lateness decreases by 18 % when the myopic algorithm is used. For all instances with base instances 1, 2, 4, 6, 7, and 9, there is no difference concerning lateness no matter if the model with optional rules or the myopic algorithm is chosen as solution technique. But the cumulated lateness over all instances is 83 % worse if the algorithm instead of the MILP model with the consideration of optional rules is used to determine the driver schedule. This shows that the MILP model with consideration of the optional rules has a significant higher potential for keeping lateness low or avoiding it. The reason for this is that by the use of the MILP models a global optimum over all stops is obtained. For the choice of time windows and the determination of driver activities the whole tour is considered. In contrast, the myopic algorithm plans driver activities successively not considering stops beyond the next one.

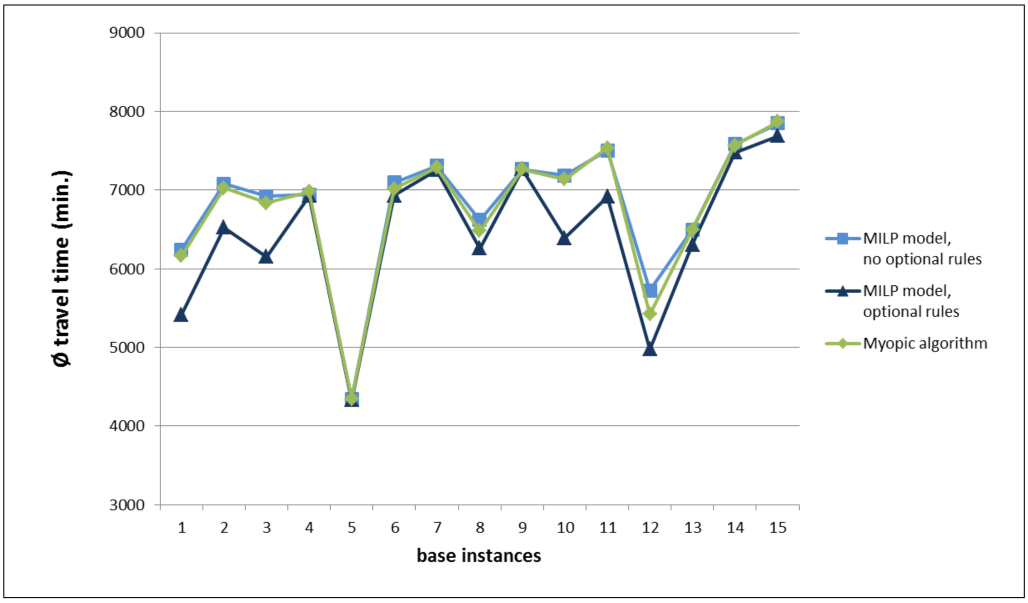

Figure 26: Average travel time of schedules depending on the solution technique

The overall travel time (see Figure 26) is on average slightly reduced (less than 1 %) if the myopic algorithm is chosen and not the model without optional rules, but compared to the model with optional rules, the schedules constructed with the myopic algorithm require on average 5 % more time. This is significant as discussed in Section 7.3.

9.2 General findings

The following figures depict schedule properties retrieved with each of the three solution techniques described in the previous sections depending on input parameters and settings.

At first, we analyze the influence of the time window properties on the overall lateness. As one might have expected, lateness tends to be less if more time windows are available. When derivating test instances from base instances, the second time window added for a chosen base instance is set to start after a time interval that follows the first time window. In the driver schedules obtained for those instances, lateness is significantly less compared to the corresponding schedules for the instances with only one time window.[1] When the third time window at a customer location is added, which ends before the start of first time window assigned, still a reduction can be noticed in all but one test set (see Figure 27). As

[1] For the derivation of test instances from base instances and the assignment of time windows see Section 7.1.

the myopic algorithm is a heuristic that only identifies local optima, i.e. an optimum for the current arc, but not the global one, deterioration is possible when time windows are added. Neglecting the possibility to use waiting time to start a daily rest period or break and choosing a time window earlier even if this causes lateness can lead to more lateness at subsequent customer locations. As expected, lateness is reduced if time windows are extended leaving their starting time constant. If two or more time windows are available with 10 hours length (simulation of opening hours), no lateness is observed at all no matter which solution technique was chosen.

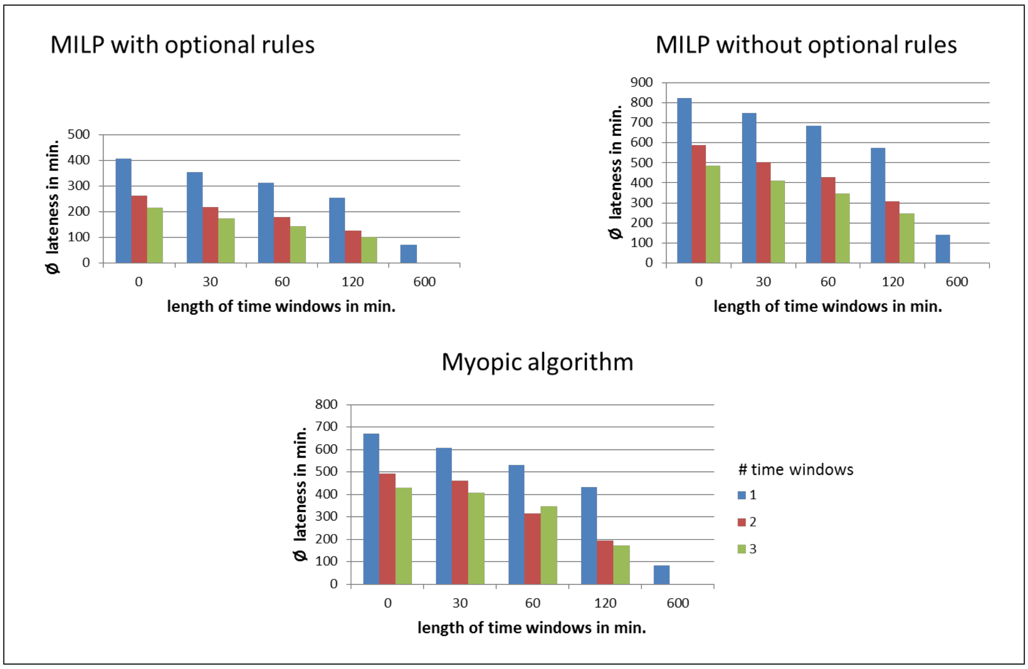

Figure 27: Impact of time windows on lateness

The overall travel time does not behave as uniformly as the lateness, since it is only considered as a subordinate optimization criterion in the MILP models. While the average overall travel time decreases with the number and the length of time windows if the model with optional rules is chosen, with the model without optional rules, the average overall travel time sometimes increases when the second time window is added. With the myopic algorithm, all average values for two time windows show this deterioration.

Figure 29 shows the average proportions of the different driver activities. It is interesting that the proportion of working time (i.e. the driving time and the time needed for loading/unloading) with 45.81 % considering the model with optional rules is reduced by about 6 % if the optional rules are neglected and by nearly 5 % if the myopic algorithm is chosen as solution technique. The proportions of rest periods, breaks and waiting time increase accordingly.

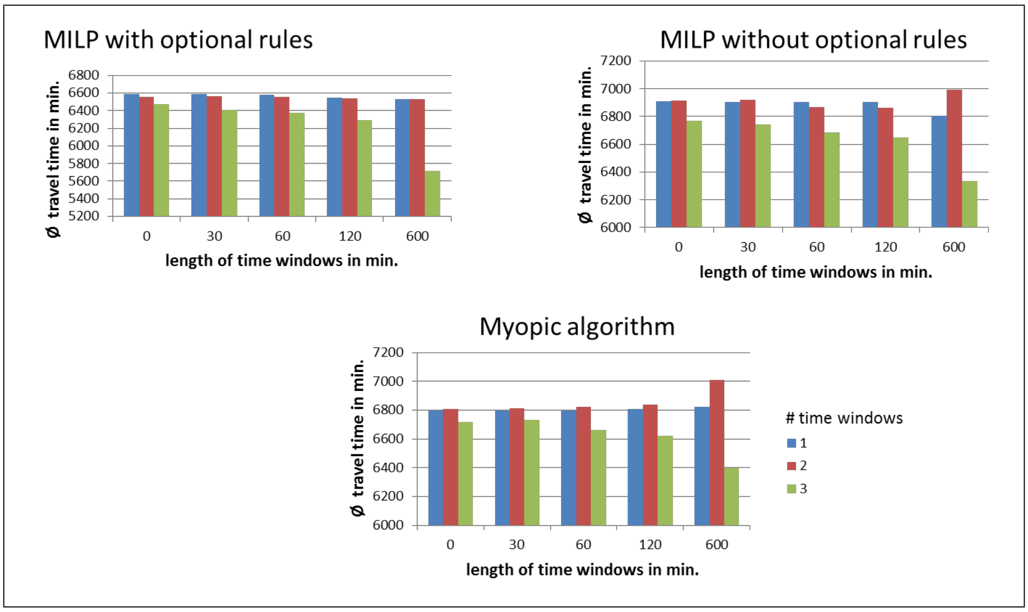

Figure 28: Time window properties and travel time

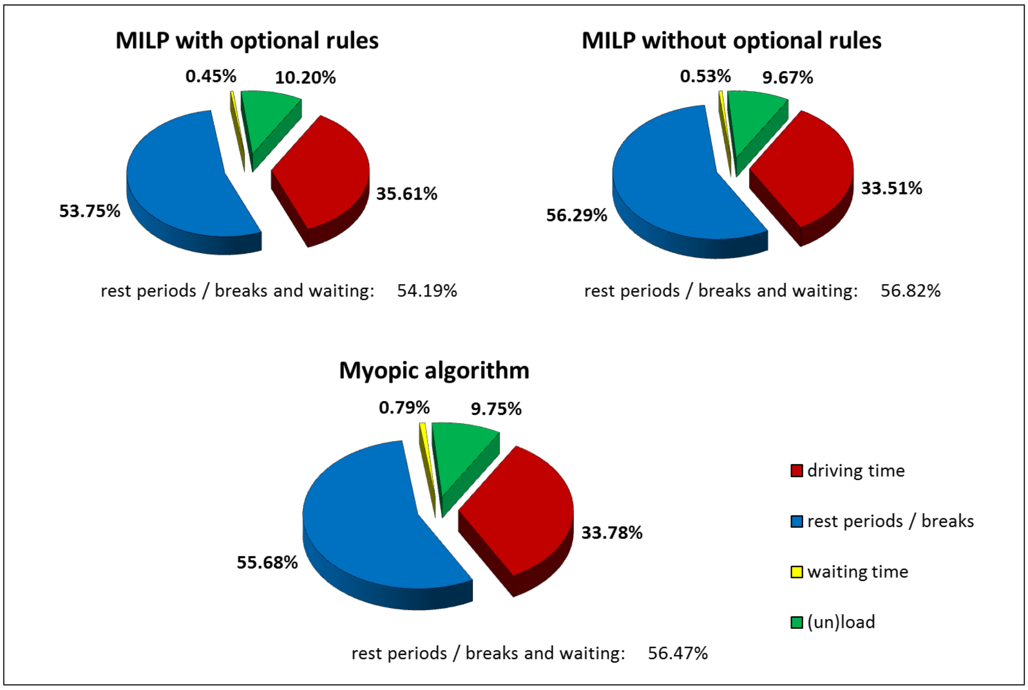

Figure 29: Proportions of different driver activities

Research has shown that long periods of night work can be harmful for the health of workers and driving at night raises safety risks for the driver himself and other participants in traffic. Therefore, Directive 2002/15/EC (European Parliament and Council of the European Union (2002)) lays down basic rules for night work that have to be implemented in national law. The rules incorporate the necessity to compensate night workers in accordance with national legislative measures. German law, i.e. the "Arbeitszeitgesetz" (Deutscher Bundestag (1994)), specifies that compensation may be a corresponding number of paid days off or an adequate surcharge on the gross remuneration. This means that the transport undertakings themselves may be interested in keeping drivers' working hours at night as low as possible.

Neither the models nor the myopic algorithm incorporate the consideration of nightly working time. Figure 30 shows the average proportions of nightly working time with no efforts taken to keep them low. For the night time we use the definition given by the "Arbeitszeitgesetz". There, the night time is defined to be the time period between 11:00 pm and 06:00 am. It is interesting to note that the average proportions of nightly working time for all solution techniques are less than the proportion of night time on the overall day with 24 hours (lower right corner of Figure 30). The reason for this may be the originally planned arrival times that served as basis for the definition of time windows. Only base instances 6, 8, 10, 13 and 14 contained arrival times which were planned during night time. Still, independent of the solution technique, the resulting working time at night is not negligible.

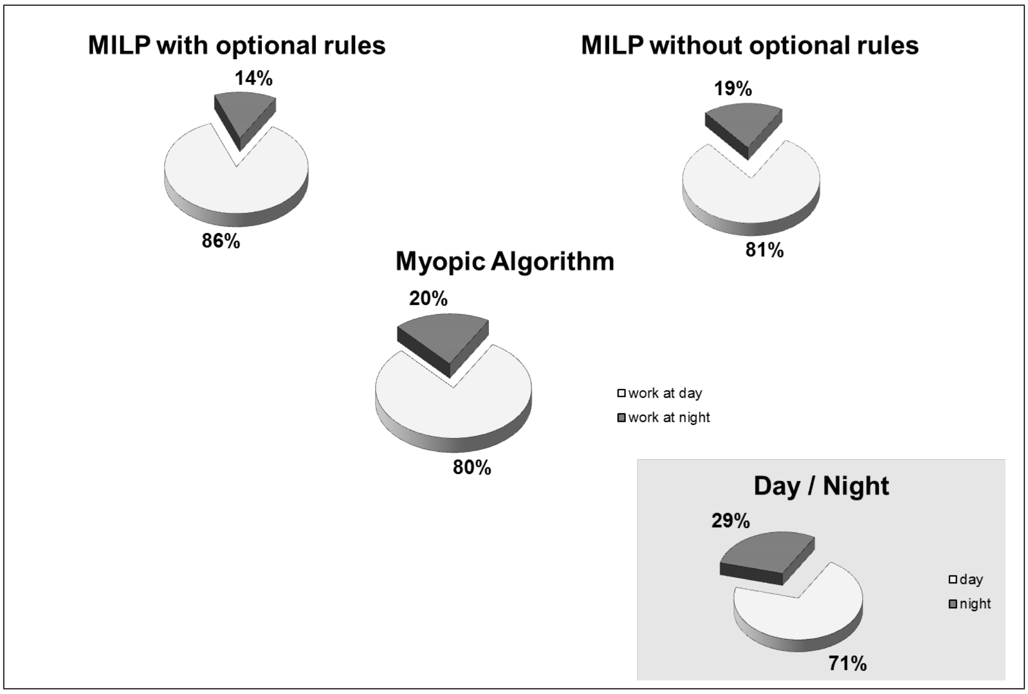

Figure 30: Proportions of working time at night and day

10 Summary and future research

In this last section, we reflect on the main results of this study and reveal aspects that are worthwile to be discussed and analyzed in future research.

10.1 Summary

Increasing just-in-time management practices, growing pressure on satisfying customer demands on time, and the need to keep transport costs low put high pressure on truck drivers, dispatchers, and their transport companies. Regulation (EC) No 561/2006 defines the rules for the number, duration and time intervals when rest periods and breaks have to be taken with the objective to ensure road safety, adequate working conditions, and undistorted competition in the road haulage sector. A strict adherence to the rules is necessary, since depending on the seriousness, infringements may have severe consequences for the drivers and the transport company itself. When determining driver schedules, the EU legislation has to be taken into account, as it has a high influence on the arrival times at customer locations and on the travel durations in general. Technology such as on-board computers, digital tachographs and telematics equipment add new challenges but also offer new planning possibilities.

The present study proposes two MILP models and optimization strategies that, together with a transformation algorithm (see Appendix A), allow to plan driver activities in compliance with Regulation (EC) No 561/2006 for a given sequence of customer locations and other stops to be visited. Each customer location has one or multiple time windows among which a choice has to be made. The incorporation of daily rest periods and breaks allows for greater planning reliability. A special feature is the consideration of "soft" time windows which has not been studied in this context so far. By penalizing lateness in the objective function instead of prohibiting the arrival outside the time windows, schedules are found even if lateness cannot be avoided. The resulting schedule gives important information to the dispatcher that is necessary to set up a better schedule. In online re-planning, lateness can be revealed at an early stage such that it is possible to reorganize the schedule or to negotiate arrival times with customers before communication effort and costs increase and further delays or cancellations are unavoidable. In this way, transport undertakings as well as their customers can benefit and pressure on dispatchers and drivers can be reduced. Attempts to choose time windows that are not reachable can be avoided.

Test instances were derived from real data provided by a German haulage company that operates vehicles in Europe. Vehicle routes were reconstructed for one week, involving between 2 and 10 customer locations and stops for start and end locations (i.e. 4 to 12 stops). Arrival times planned by the dispatchers were used as a basis to generate different time windows. The number of time windows and their length were varied to obtain different test instances. We examined the run time, lateness and overall travel time for all of the instances depending on the number of stops, and the number and length of time windows. For all of the instances, reasonable run times were achieved ranging from 0.03 seconds (4 stops, 3 time windows, time window length: 10 h) to 10.94 seconds (11 stops, 3 time windows, time window length: 10 h) for the MILP model with consideration of the optional rules on a desktop computer, where the number of stops had the most influence on the run time.

The optional rules were deactivated in the original MILP model to test their influence on the above criteria. The run time was reduced significantly by between 37 % (base instance 1, 4 stops) up to 81 % (base instance 15, 12 stops). On the other hand, the overall lateness was 55 % less if the optional rules were allowed and the overall travel time was reduced by 5 %.

The proposed MILP models allow to establish an optimal driver schedule with the help of optimization software. As the use of a commercial solver can be an obstacle for a company due to cost reasons, we wished to investigate the magnitude of the schedule improvement compared to a heuristic that should simulate the usage of sophisticated strategies by a dispatcher. Based on the idea of a driver status that is modified with each new activity, a myopic algorithm was developed that can only "see" the route until the next customer stop and the corresponding customer time window in advance, and plans driver activities accordingly. Simple strategies were chosen to also integrate the optional rules.

The myopic algorithm achieved 18 % overall lateness reduction and no increase in the average overall travel time in comparison with the model without optional rules. Together with the run time of the myopic algorithm that was less than 1 millisecond, the algorithm itself is interesting. The main advantages of the myopic algorithm are its short run time and that no optimization solver is necessary to obtain a solution. The short-sightedness and concentration on one arc at a time makes resulting schedules easy to understand. Similar to the models, the possibility to start with a given driver status allows for online re-planning.

The consideration of the complete tour with all stops allows to construct a schedule with globally optimized lateness (the most important criterion) and overall travel time when solving the MILP model with optional rules. The overall lateness was 45 % less compared to the myopic algorithm and the average overall travel time was reduced by 3.5 %. The run time is longer, but depending on the fleet size, the length of the planning horizon and the available computing capacity, online re-planning still may be considered.

The largest advantage of the MILP models is the determination of a global optimum over all stops. If the customer locations to be visited in the considered week are not known in advance but only for the next one or two stops, the dispatcher has to choose among different opportunities without exactly knowing future requests. This reduces the benefit of the MILP model with optional rules.

10.2 Future research

The basis for planning of vehicle routes should be reliable driving durations that consider various traffic conditions that are dependent on the routes traveled and the time of the day (see, for example, Kok (2010)). Travel times may vary significantly as there are differences, for example, between traveling on a Saturday, in rush-hour traffic or at the start of vacations. Even though online re-planning is possible, when scheduling driver activities, the estimated driving duration should consider time buffers to compensate for delays due to unexpected events such as traffic jams or detours because of blocked roads. To find reasonable time buffers when scheduling customer stops and driver activities is worth further consideration. The robustness depending on the measures chosen could be analyzed by simulating deteriorations and using online re-planning with a lesser degree of freedom to determine modified schedules.

The presented techniques to plan driver activities do not consider the location of possible rest areas. Especially when a daily rest period has to be taken, drivers often face the problem of finding an adequate location. Even though modern parking guidance systems are available at some locations, nowadays rest areas are often overcrowded. Depending on the time of the day, drivers often have to search intensively for a place to spend their daily rest period. Goel (2012) proposes an approach that only allows breaks and rest periods at rest areas. It is worth taking a closer look at the integration of information about rest areas into the model with multiple soft time windows. Additionally, detours to reach rest areas could also be considered.

The integration of the rules of Directive 2002/15/EC (European Parliament and Council of the European Union (2002)) should be studied. This comprises the rule that working time has to be interrupted by a break of at least 30 minutes if the sum of all working hours is between six and nine hours and of at least 45 minutes if the driver works more than nine hours. Furthermore, Directive 2002/15/EC contains a framework to define rules for night work. If night work is performed, the daily working time is not allowed to exceed ten hours in each 24 hours period and it has to be compensated by the employer. Different implementations in national laws do exist. The numerical experiments in Section 9.2 show that a significant part of the overall working time can be at night if no measures are taken to keep it low. Depending on the strategy of a haulage company it would, for example, be possible to define time intervals in which no work is allowed. Another approach would be to allow not more than 10 hours working time between two daily rest periods and to introduce a cost function for working time at night.

Especially when considering long-haul trips, driving bans on public holidays need to be integrated into driver scheduling. As there are regional differences, the integration into the combined problem of route planning (or vehicle routing) and driver scheduling seems reasonable.

Truck drivers face difficult working conditions. In long-haul international transport, truck drivers spend long periods on-road away from home. Competitors and client demands such as just-in-time management induce high pressure. Moreover, remote monitoring and complex technology act as a deterrent because drivers may feel permanently observed and monitored or demoralized. Accessibility of facilities and services (hygienic, food and medical) is not always the best and road safety risks are not negligible. All these reasons lead to a low attractiveness of the profession and a shortage of qualified drivers (European Commission (2014)). Besides lowering pressure by employing better and more realistic planning techniques, an improvement on working conditions can be achieved. Planning techniques that take into account different amenities at resting places where drivers take their daily rest periods and breaks would help to raise the attractiveness of the profession.

Appendices

A Transformation to a driver schedule

The following algorithm transforms the solution of a MILP model (the model with consideration of the optional rules or the model without) into a uniquely determined detailed driver schedule. The method $addActivity\,((i, i+1), \langle type \rangle, \langle duration \rangle)$ adds an activity of type $\langle type \rangle$ and duration $\langle duration \rangle$ as last activity on arc $(i, i+1)$ that in this case represents the time interval between the start after unloading/loading at customer i and the end of loading/unloading at customer $i+1$.

$restCarryover_1$: If there is much time to reach a time window, it is possible that a rest on an arc is planned that lasts longer than 11 hours and another rest is planned in the subsequent vertex. In that case, the difference of the duration of the last rest on the arc and 11 hours is added to the duration of the rest taken in the vertex.

$restCarryover_2$: It is possible that the solution has one rest more than actually needed to traverse the arc. To prevent two consecutive rest periods without another activity in between, this rest is mapped on the subsequent vertex.

$helpPartialRest$: Indicates if a partial rest is taken on the arc and not in vertex $i+1$.

Algorithm 1 Computing a Driver Schedule

Input: model solution

Output: a list of driver activities

```
// Initialize
restCarryover_1 ← 0
restCarryover_2 ← 0
helpPartialRest ← 0
ptwr ← Time at the start of the schedule
duration ← 0

// ————————————————————————————————
// Determine activities in the first vertex
// ————————————————————————————————
if α★_0^rest = 1 then
    duration ← Δ★_0^rest
```

$\quad addActivity\,((0,1), rest, duration)$
$\quad ptwr \leftarrow ptwr + duration$
else if $\overset{\star}{\alpha}{}_0^{break} = 1$ **then**
$\quad$ **if** $\overline{hpb} = 1$ **then**
$\quad\quad duration \leftarrow \max\left(0, 30 - \overline{ubt}\right)$
$\quad$ **else**
$\quad\quad duration \leftarrow \max\left(0, 45 - \overline{ubt}\right)$
$\quad$ **end if**
$\quad addActivity\,((0,1), rest, duration)$
$\quad ptwr \leftarrow ptwr + duration$
else if $\overset{\star}{\alpha}{}_0^{pbreak} = 1$ **then**
$\quad duration \leftarrow \max\left(0, 15 - \overline{ubt}\right)$
$\quad addActivity\,((0,1), rest, duration)$
$\quad ptwr \leftarrow ptwr + duration$
else if $\overset{\star}{\alpha}{}_0^{prest} = 1$ **then**
$\quad duration \leftarrow \max(0, 180 - \overline{urt})$
$\quad addActivity\,((0,1), rest, duration)$
$\quad ptwr \leftarrow ptwr + duration$
end if
if $\overset{\star}{\Delta}{}_0^{wait} > 0$ **then**
$\quad duration \leftarrow \overset{\star}{\Delta}{}_0^{wait}$
$\quad addActivity\,((0,1), wait, duration)$
$\quad ptwr \leftarrow ptwr + duration$
end if

// ————————————————————————————————
// Calculate activities "between" customer locations i and $i+1$
// ————————————————————————————————
for $i = 0$ **to** $n-1$ **do**

$\quad$ // Use driving time left until next break or rest period to partially or
$\quad$ // completely traverse the arc $(i, i+1)$.
$\quad$ **if** $\overset{\star}{L}{}_i^{dt} > 0$ **then**
$\quad\quad$ **if** $\bar{\Delta}_{(i,i+1)}^{drive} > 0$ **then**

$duration \leftarrow \min\left(\overset{\star}{L}{}_i^{dt}, \bar{\Delta}^{drive}_{(i,i+1)}\right)$
$addActivity\,((0,1), drive, duration)$
$ptwr \leftarrow ptwr + duration$
end if
end if

// No early daily rest period is planned as first
// resting activity on arc $(i, i+1)$.
if $\overset{\star}{\mu}{}^{earlydr1}_{(i,i+1)} = 0$ **then**

// The distance between i and $i+1$ is greater than the
// driving time left until the next break or rest period.
if $\bar{\Delta}^{drive}_{(i,i+1)} > \overset{\star}{L}{}_i^{dt}$ **then**

// Take a break if the daily driving time left is greater than
// the driving time left until the next break. Afterwards, continue
// driving until the daily driving time reaches its limit or customer
// $i+1$ is reached.
if $\overset{\star}{L}{}_i^{ddt} > \overset{\star}{L}{}_i^{dt}$ **then**
if $\overset{\star}{l}{}_i^{pbreak} = 1$ **then**
$duration \leftarrow 30$
else
if $\overset{\star}{\alpha}{}^{prest}_{i+1} = 1 \wedge \bar{\Delta}^{drive}_{(i,i+1)} < \overset{\star}{L}{}_i^{ddt} \wedge \overset{\star}{\mu}{}^{extd1}_{(i,i+1)} = 0 \wedge \overset{\star}{\mu}{}^{extd}_{i+1} = 0$ **then**
$duration \leftarrow 180$
$helpPartialRest \leftarrow 1$
else
$duration \leftarrow 45$
end if
end if
$addActivity\,((i, i+1), rest, duration)$
$ptwr \leftarrow ptwr + duration$

$duration \leftarrow \min\left(\overset{\star}{L}{}_i^{ddt} - \overset{\star}{L}{}_i^{dt}, \bar{\Delta}^{drive}_{(i,i+1)} - \overset{\star}{L}{}_i^{dt}\right)$
$addActivity\,((i, i+1), drive, duration)$

$ptwr \leftarrow ptwr + duration$
end if

// If a driving time extension of type 1 is used, take an additional
// break or first partial daily rest period.
if $\overset{\star}{\mu}^{extd1}_{(i,i+1)} = 1$ **then**
if $\overset{\star}{\alpha}^{prest}_{i+1} = 1 \wedge \overset{\star}{\mu}^{extd}_{i+1} = 0 \wedge \overset{\star}{\Delta}^{rest}_{(i,i+1)} = 0$ **then**
$duration \leftarrow 180$
$helpPartialRest \leftarrow 1$
else
$duration \leftarrow 45$
end if
$addActivity\,((i, i+1), rest, duration)$
$ptwr \leftarrow ptwr + duration$

// If the time suffices, 60 minutes of driving do follow, otherwise
// the remaining time left until the next rest period is exploited.
if $\overset{\star}{\lambda}^{5}_{i} = 1$ **then**
$duration \leftarrow 60$
else
if $\overset{\star}{l}^{pbreak}_{i} = 1$ **then**
$duration \leftarrow \overset{\star}{L}^{t}_{i} - \overset{\star}{L}^{ddt}_{i} - 30$
else
$duration \leftarrow \overset{\star}{L}^{t}_{i} - \overset{\star}{L}^{ddt}_{i} - 45$
end if
if $\overset{\star}{L}^{ddt}_{i} > \overset{\star}{L}^{dt}_{i}$ **then**
$duration \leftarrow duration - 45$
end if
end if
$duration \leftarrow \min\left(duration, \bar{\Delta}^{drive}_{(i,i+1)} - \overset{\star}{L}^{ddt}_{i}\right)$
$addActivity\,((i, i+1), drive, duration)$
$ptwr \leftarrow ptwr + duration$
end if
end if

end if

// ————————————————————————————————
// If at least one daily rest period should be made on arc $(i, i+1)$,
// schedule now the first daily rest period.
// ————————————————————————————————
$restCarryover_1 \leftarrow \overset{\star\,rest}{\Delta}_{(i,i+1)} - \overset{\star\,rest}{A}_{(i,i+1)} \cdot 660 + 120 \cdot \overset{\star\,redrest}{\mu}_{(i,i+1)} + 120 \cdot \overset{\star prest}{l_i}$
if $\overset{\star\,rest}{A}_{(i,i+1)} \geq 1$ **then**
  // In case that $\overset{\star\,ddt}{E}_{i+1}$ is equal to 540, the rest period can be
  // postponed to the subsequent vertex. Only take a daily rest period
  // in the opposite case.
  if $\overset{\star\,ddt}{E}_{i+1} < 540$ **then**
    if $\overset{\star\,redrest}{\mu}_{(i,i+1)} = \overset{\star\,rest}{A}_{(i,i+1)} \vee \overset{\star prest}{l_i} = 1 \vee \overset{\star dredrest}{l_i} = 1$ **then**
      $duration \leftarrow 540$
    else
      $duration \leftarrow 660$
    end if
    if $\overset{\star\,rest}{\alpha}_{i+1} = 0 \wedge \overset{\star\,rest}{A}_{(i,i+1)} = 1$ **then**
      $duration \leftarrow duration + restCarryover_1$
    end if
    $addActivity\,((i, i+1), rest, duration)$
    $ptwr \leftarrow ptwr + duration$
  end if
end if
// ————————————————————————————————
// Plan driver activities between the first and the last daily rest period
// on arc $(i, i+1)$
// ————————————————————————————————
for $k = \overset{\star\,rest}{A}_{(i,i+1)}$ **to** 2 **do**
  $duration \leftarrow 270$
  $addActivity\,((i, i+1), drive, duration)$
  $ptwr \leftarrow ptwr + duration$
  if $k > 2 \vee \overset{\star\,earlydr2}{\mu}_{(i,i+1)} = 0$ **then**
    $duration \leftarrow 45$

$addActivity\,((i, i+1), rest, duration)$
$ptwr \leftarrow ptwr + duration$

$duration \leftarrow 270$
$addActivity\,((i, i+1), drive, duration)$
$ptwr \leftarrow ptwr + duration$

// Schedule extended driving times of type 2 as late as possible.
if $k \leq \overset{\star}{\mu}^{extd2}_{(i,i+1)} + 1$ **then**
$duration = 45$
$addActivity\,((i, i+1), rest, duration)$
$ptwr \leftarrow ptwr + duration$

$duration \leftarrow 60$
$addActivity\,((i, i+1), drive, duration)$
$ptwr \leftarrow ptwr + duration$
end if
if $k > 2$ **then**
// Schedule reducd daily rest periods as late as possible.
// In case that $\overset{\star}{l}^{dredrest}_{i} = 1$, one reduced daily rest period
// has to be the first daily rest period on this arc.
if $k \leq \overset{\star}{\mu}^{redrest}_{(i,i+1)} - \overset{\star}{l}^{dredrest}_{i}$ **then**
$duration \leftarrow 540$
$addActivity\,((i, i+1), rest, duration)$
$ptwr \leftarrow ptwr + duration$
else
$duration \leftarrow 660$
$addActivity\,((i, i+1), rest, duration)$
$ptwr \leftarrow ptwr + duration$
end if
end if
end if
$k \leftarrow k - 1$
end for
// ———————————————

// Plan last daily rest period in case that more than one daily rest period
// is taken on arc $(i, i+1)$.
// ————————————————————————————————
if $\overset{\star}{A}^{rest}_{(i,i+1)} \geq 2$ **then**
  if $\overset{\star}{E}^{ddt}_{i+1} < 540$ **then**
    if $\overset{\star}{\mu}^{redrest}_{(i,i+1)} \geq 1 + \overset{\star}{l}^{prest}_{i} + \overset{\star}{l}^{dredrest}_{i}$ **then**
      $duration \leftarrow 540$
    else
      $duration \leftarrow 660$
    end if
    if $\overset{\star}{\alpha}^{rest}_{i+1} = 0$ **then**
      $duration \leftarrow duration + restCarryover_1$
    end if
    $addActivity\left((i, i+1), rest, duration\right)$
    $ptwr \leftarrow ptwr + duration$
  end if
end if
// ————————————————————————————————
// Plan driver activities after the last daily rest period on arc $(i, i+1)$.
// ————————————————————————————————
if $\overset{\star}{A}^{rest}_{(i,i+1)} \geq 1$ **then**
  if $\overset{\star}{E}^{ddt}_{i+1} < 540$ **then**
    $duration \leftarrow \min\left(270, 540 - \overset{\star}{E}^{ddt}_{i+1}\right)$
    $addActivity\left((i, i+1), drive, duration\right)$
    $ptwr \leftarrow ptwr + duration$
    if $\overset{\star}{E}^{ddt}_{i+1} < 270$ **then**
      if $\overset{\star}{\alpha}^{prest}_{i+1} = 1 \wedge \overset{\star}{\mu}^{extd}_{i+1} = 0 \wedge \overset{\star}{\mu}^{extd3}_{(i,i+1)} = 0$ **then**
        $duration \leftarrow 180$
        $helpPartialRest \leftarrow 1$
      else
        $duration \leftarrow 45$
      end if
      $addActivity\left((i, i+1), rest, duration\right)$

$ptwr \leftarrow ptwr + duration$

if $\overset{\star}{\mu}^{extd3}_{(i,i+1)} = 0$ **then**

$duration \leftarrow 270 - \overset{\star}{E}^{ddt}_{i+1}$

$addActivity\,((i, i+1), drive, duration)$

$ptwr \leftarrow ptwr + duration$

else

$duration \leftarrow 270$

$addActivity\,((i, i+1), drive, duration)$

$ptwr \leftarrow ptwr + duration$

if $\overset{\star}{\alpha}^{prest}_{i+1} = 1 \wedge \overset{\star}{\mu}^{extd}_{i+1} = 0$ **then**

$duration \leftarrow 180$

$helpPartialRest \leftarrow 1$

else

$duration \leftarrow 45$

end if

$addActivity\,((i, i+1), rest, duration)$

$ptwr \leftarrow ptwr + duration$

$duration \leftarrow 60 - \overset{\star}{E}^{ddt}_{i+1}$

if $duration > 0$ **then**

$addActivity\,((i, i+1), drive, duration)$

$ptwr \leftarrow ptwr + duration$

end if

end if

end if

end if

end if

// ——————————————————————————————

// Plan driver activities at arrival at customer $i+1$.

// ——————————————————————————————

// If a daily rest period is planned for arc $(i, i+1)$ that can be postponed

// to the subsequent vertex, do it. If a daily rest period is also scheduled

// at customer location $i+1$, unite the two daily rest periods.

if $\overset{\star}{E}^{ddt}_{i+1} = 540 \wedge \overset{\star}{A}^{rest}_{(i,i+1)} \geq 1$ **then**

if $\overset{\star redrest}{\mu}_{(i,i+1)} \geq 1 + \overset{\star prest}{l_i} + \overset{\star dredrest}{l_i}$ **then**
$\quad\quad restCarryover_2 \leftarrow 540 + restCarryover_1$
else
$\quad\quad restCarryover_2 \leftarrow 660 + restCarryover_1$
end if
end if
if $\overset{\star rest}{\alpha}_{i+1} = 1$ **then**
$\quad duration \leftarrow \overset{\star rest}{\Delta}_{i+1}$
if $\overset{\star rest}{A}_{(i,i+1)} > 0$ **then**
$\quad\quad duration \leftarrow duration + restCarryover_1$
end if
$\quad addActivity\left((i, i+1), rest, duration\right)$
$\quad ptwr \leftarrow ptwr + duration$
else if $restCarryover_2 > 0$ **then**
$\quad duration \leftarrow restCarryover_2$
$\quad addActivity\left((i, i+1), rest, duration\right)$
$\quad ptwr \leftarrow ptwr + duration$
else if $\overset{\star break}{A}_{(i,i+1)} \geq 1 \wedge \overset{\star dt}{E}_{i+1} = 270 \wedge \overset{\star ddt}{E}_{i+1} \leq 270$ **then**
if $\overset{\star pbreak}{l_i} = 1 \wedge \overset{\star rest}{A}_{(i,i+1)} = 0$ **then**
$\quad\quad duration \leftarrow 30$
else
$\quad\quad duration \leftarrow 45$
end if
$\quad addActivity\left((i, i+1), rest, duration\right)$
$\quad ptwr \leftarrow ptwr + duration$
end if
if $\overset{\star break}{\alpha}_{i+1} = 1$ **then**
if $\overset{\star pbreak}{l_i} = 1 \wedge \overset{\star break}{\alpha}_{(i,i+1)} = 0$ **then**
$\quad\quad duration \leftarrow 30$
else
$\quad\quad duration \leftarrow 45$
end if
$\quad addActivity\left((i, i+1), rest, duration\right)$
$\quad ptwr \leftarrow ptwr + duration$

else if $\overset{\star}{\alpha}^{pbreak}_{i+1} = 1$ **then**
$duration \leftarrow 15$
$addActivity\,((i, i+1), rest, duration)$
$ptwr \leftarrow ptwr + duration$
else if $\overset{\star}{\alpha}^{prest}_{i+1} = 1 \wedge helpPartialRest = 0$ **then**
$duration \leftarrow 180$
$addActivity\,((i, i+1), rest, duration)$
$ptwr \leftarrow ptwr + duration$
end if
if $\overset{\star}{\Delta}^{wait}_{i+1} > 0$ **then**
$duration \leftarrow \overset{\star}{\Delta}^{wait}_{i+1}$
$addActivity\,((i, i+1), wait, duration)$
$ptwr \leftarrow ptwr + duration$
end if
if $i < r - 1 \wedge \bar{\Delta}^{service}_{i+1} > 0$ **then**
$duration \leftarrow \bar{\Delta}^{service}_{i+1}$
$addActivity\,((i, i+1), work, duration)$
$ptwr \leftarrow ptwr + duration$
end if
$i \leftarrow i + 1$
end for

B Myopic algorithm - Pseudo-code

At the beginning of the planning horizon, the driver status is given by the tupel

$$driverStatus = (\overline{ptwr}, \overline{udt}, \overline{ddt}, \overline{ptr}, \overline{hpb}, \overline{hpr}, \overline{noRed}, \overline{noExt}, \overline{red}, \overline{dte}).$$

The corresponding status variables are initialized in Algorithm 2.

Algorithm 2 Initialize driver status

// Initialize: Set starting driver status

$udt \leftarrow \overline{udt}$
$ddt \leftarrow \overline{ddt}$
$ptr \leftarrow \overline{ptr}$

$ptwr \leftarrow \overline{ptwr}$
$hpb \leftarrow \overline{hpb}$
$hpr \leftarrow \overline{hpr}$
$noRed \leftarrow \overline{noRed}$
$noExt \leftarrow \overline{noExt}$
$dte \leftarrow 0$
$red \leftarrow 0$

// If the driving time since the last daily or weekly rest period exceeds 9 hours, a
// driving time extension is active.

if $ddt > 540 \land noExt <= 2$ **then**
  $dte \leftarrow 1$
end if

// If the time since the last daily or weekly rest period exceeds 13 hours, plan that
// the next daily rest period has to be a reduced one.

if $ptr > 780 \land noRed < 3$ **then**
  $red \leftarrow 1$
end if

The method *scheduleActivity* (<last stop>, <duration in min.>, <activity type>) schedules an activity with given activity type *activityType* and duration *duration*, i.e. adds it at the end of the list of activities between stops i and $i+1$. The update of the driver status is done accordingly and the algorithm used can be seen as an extension to optional rules of the label-update made in the labeling algorithms of Goel (2009). The pseudo-code is given by Algorithm 3.

Algorithm 3 Update driver status

---

switch $activityType$

  $ptwr \leftarrow ptwr + duration$

  case $drive$
    $udt \leftarrow udt + duration$
    $ddt \leftarrow ddt + duration$
    $ptr \leftarrow ptr + duration$
    break
  end case

  case $work$
    $ptr \leftarrow ptr + duration$
    break
  end case

```
    case wait
        ptr ← ptr + duration
        break
    end case

    case break
        ptr ← ptr + duration
        if duration > 15 then
            udt ← 0
            hpb ← 0
        else
            hpb ← 1
        end if
        break
    end case

    case rest
        udt ← 0
        if duration = 180 then
            hpr ← 1
            ptr ← ptr + 180
        else
            ddt ← 0
            ptr ← 0
            hpb ← 0
            hpr ← 0
            red ← 0
            dte ← 0
        end if
        break
    end case

    case redrest
        udt ← 0
        ddt ← 0
        ptr ← 0
        hpb ← 0
        hpr ← 0
        red ← 0
        dte ← 0
        noRed ← noRed + 1
        break
    end case
end switch
```

Algorithm 4 is used, to determine the first reachable time window. This is for example needed in Algorithm 5 if a driving time extension is considered or in Algorithm 6 right at the beginning.

Algorithm 4 Determine first reachable time window

Input: last stop i, potential arrival time $time$ at customer location $i+1$, time window information for $i+1$ (next stop)
Output: first time window reachable without lateness ignoring a potential daily rest period still to take or last time window if lateness is not avoidable

$z \leftarrow 0$
$minLateness \leftarrow \max\left(time - \overline{TW}^{end}_{i+1,0}, 0\right)$

for $k = 1$ **to** $nbTW_i - 1$ **do**
  $lateness \leftarrow \max\left(time - \overline{TW}^{end}_{i+1,k}, 0\right)$
  if $\left(\begin{array}{l}(lateness < minLateness) \\ \vee(time \leq \overline{TW}^{end}_{i+1,k} \wedge \overline{TW}^{begin}_{i+1,k} < \overline{TW}^{begin}_{i+1,z})\end{array}\right)$ **then**
    $minLateness = lateness$
    $z \leftarrow k$
  end if
end for
return z

For each pair of consecutive stops i and $i+1$ Algorithms 5, 6, and 7 are executed one after another to determine the driver schedule.

Algorithm 5 Schedule activities on arc (i,i+1)

// ————————————————————————
// Schedule activities "between" stops i and $i+1$.
// (Durations of daily rest periods may be modified later.)
// ————————————————————————

$duration \leftarrow 0$
$drivingTimeToDest \leftarrow \bar{\Delta}^{drive}_{(i,i+1)}$

while $drivingTimeToDest > 0$ **do**

  // Determine the next driving time interval as the minimum of the nonstop
  // driving time left, the daily driving time left, the time until the next daily rest
  // period and the driving time still needed to reach the next stop.
  // If a partial daily rest period was made or it was decided previously that the
  // next daily rest period will be a reduced one, add two hours to the time until
  // the next daily rest period has to start.

$$duration \leftarrow \min \begin{pmatrix} 270 - udt, \\ 540 + 60\ dte - ddt, \\ 780 + 120\ hpr + 120\ red - ptr, \\ drivingTimeToDest \end{pmatrix}$$

$scheduleActivity\,(i, duration, drive)$

Update driver status

$drivingTimeToDest \leftarrow drivingTimeToDest - duration$

if $drivingTimeToDest = 0$ **then**
  break
else

  // ————————————————————————————————
  // If less than one hour of driving is left to reach the next stop, take a driving
  // time extension if possible and advantageous.
  // ————————————————————————————————

  // A driving time extension is considered
  // - if at most one hour of driving is left until the next stop is reached,
  // - if at least one daily rest period on the current arc has already been made,
  // - if in the current week, less than two driving time extensions have been taken
  // - if it is possible to save a daily rest period on the current arc and thus
  // - reach an earlier time window or reduce lateness.

  if $\begin{pmatrix} (drivingTimeToDest \leq 60) \\ \wedge\ (getDailyRestPosSize() > 0) \\ \wedge\ (ddt = 540) \\ \wedge\ (noExt < 2) \end{pmatrix}$ **then**

    $redRestPoss \leftarrow 0$

    if $noRed < 3$ **then**
      $redRestPoss \leftarrow 1$
    end if

    if $\begin{pmatrix} 780 + 120\ redRestPoss \\ \quad \geq ptr + drivingTimeToDest + 45 + workingTime \end{pmatrix}$ **then**

      $time1 \leftarrow ptwr + 45 + drivingTimeToDest$
      $time2 \leftarrow ptwr + 660 + drivingTimeToDest$

      // Determine the first reachable time window for both alternatives
      // (Algorithm 4).

      $z1 \leftarrow$ **Determine first reachable time window** for $time1$

```
        $z2 \leftarrow$ **Determine first reachable time window** for $time2$

        // If the first reachable time window starts earlier or the lateness is less if
        // a driving time extension is used, take the driving time extension.

        **if** $\overline{TW}^{begin}_{i+1,z1} < \overline{TW}^{begin}_{i+1,z2} \vee time2 - \overline{TW}^{end}_{i+1,z2} > 0$ **then**

          $scheduleActivity\,(i, 45, break)$
          **Update driver status**

          $scheduleActivity\,(i, drivingTimeToDest, drive)$
          **Update driver status**

          $dte \leftarrow 1$
          $noExt \leftarrow noExt + 1$

          **if** $780 < ptr + \bar{\Delta}^{service}_{i+1}$ **then**
            $red = 1$
          **end if** // if $780 < ptr + \bar{\Delta}^{service}_{i+1}$
          **break** // Leave while loop
        **end if** // if $\overline{TW}^{begin}_{i+1,z1} < \overline{TW}^{begin}_{i+1,z2} \vee time2 - \overline{TW}^{end}_{i+1,z2} > 0$
      **end if** // $780 + 120\ redRestPoss\ \geq ptr + \ldots$
    **end if** // if $drivingTimeToDest \leq 60 \ldots$

    // If the daily driving time or the time until the next daily rest period is
    // exhausted, take a daily rest period. Otherwise, the nonstop driving time
    // equals 4.5 hours and a break has to be taken.

    **if** $ddt = 780 \vee ptr + 45 - 15\ hpb \geq 780$ **then**
      $duration \leftarrow 660 - 120\ (hpr + red)$
      **if** $red = 1$ **then**
        $scheduleActivity\,(i, duration, redrest)$
      **else**
        $scheduleActivity\,(i, duration, rest)$
      **end if**
      **Update driver status**
    **else**
      $duration \leftarrow 45 - 15\ hpb$
      $scheduleActivity\,(i, duration, rest)$
      **Update driver status**
    **end if** // if $ddt = 780 \vee ptr + 45 - 15\ hpb \geq 780$
  **end if** // if $drivingTimeToDest = 0$
**end while** // while $drivingTimeToDest > 0$
```

As we do include standard rule 6, we have to ensure that the maximum time interval between two (daily) rest periods is not exceeded when waiting time and time for loading/unloading is scheduled. Algorithm 6 schedules an additional daily rest period if necessary. To avoid lateness, the duration of daily rest periods between stops i and $i+1$ may be reduced. If avoidance is not possible, it is tried to remove the last daily rest period by extending the duration of the previous daily rest period to compensate waiting time. The method $extendRestDurationLastRest\,(< modifier\ in\ min. >)$ in Algorithm 6 extends the duration of the last scheduled daily rest period and modifies $ptwr$ and the starting times of subsequent activities accordingly.

Algorithm 6 Choose time window at stop $i+1$

// ———————————————————————————————
// Determine the first reachable time window. Plan an additional daily rest period
// if it is necessary before loading or unloading may start. If this causes lateness
// for the time window currently considered, first try to reduce the duration of daily
// rest periods on this arc. If this does not work, try to leave out the last daily
// rest period:
// If waiting time occurs, compensate it if possible, by extending the previous daily
// rest period on this arc. Additionally, consider the option to plan the next daily rest
// period to be a reduced one to obtain two additional hours until the next daily
// rest period is necessary.
// ———————————————————————————————

$z \leftarrow$ **Determine first reachable time window** for $ptwr$

repeat
    $chosenTWEnd \leftarrow \overline{TW}^{end}_{i+1,z}$
    $z \leftarrow z+1$
    $waitingTime \leftarrow \max\left(0, \overline{TW}^{begin}_{i+1,z} - ptwr\right)$
    $dailyTimeAfterService \leftarrow ptr + waitingTime + \bar{\Delta}^{service}_{i+1}$

    if $dailyTimeAfterService > 780 + 120\ (\overline{hpr} + red)$ **then**

        // Without daily rest period, the time does not suffice to wait and serve the
        // customer. Try to schedule a daily rest period.

        if $(hpr = 1) \lor (red = 1)$ **then**
            $duration \leftarrow 540$
        else
            $duration \leftarrow 660$
        end if

        if $ptwr + duration > chosenTWEnd$ **then**

            // ———————————————————————————
            // If lateness occurs, test, whether reducing rest periods on the current arc
            // helps to reach the chosen time window in time.
            // ———————————————————————————

```
    // Determine the number of daily rest periods on the current arc that may be
    // reduced.

    posNoReductions ← getNoReducableRestPeriods()

    if (hpr = 0) ∧ (red = 0) then
        posNoReductions ← posNoReductions + 1
    end if

    posNoReductions = min (posNoReductions, 3 − noRed)

    if ptwr + duration − 120 posNoReductions ≤ chosenTWEnd then

        if red = 0 then
            scheduleActivity (i, duration, rest)
        else
            scheduleActivity (i, duration, redrest)
        end if

        Update driver status

        break

    else

        // If it is not possible to schedule a daily rest period without lateness, try
        // to extend the last daily rest period on this arc by the waiting time
        // to shift the 24 h hours time interval and/or try to extend it by deciding
        // that the next daily rest period should be a reduced one.

        reducedRestPoss = 0

        if (noRed < 3) ∧ (hpr = 0) then
            reducedRestPoss = 1
        end if

        hadDailyRest = 0

        if getNoReducableRestPeriods() > 0 then
            hadDailyRest = 1
        end if

        if ( (dailyTimeAfterService − hadDailyRest · waitingTime
                 ≤ 780 + 120 (hpr + reducedRestPoss)
             ∧ getNoRestPeriods() > 1 ) then

            if hadDailyRest > 0 then
```

```
            extendRestDurationLastRest (waitingTime)
         end if // if hadDailyRest > 0

         if ptr + waitingTime + Δ̄_{i+1}^{service} > 780 ∧ red = 0 then
            red ← 1
         end if

         break

      else

         // A daily rest period is necessary, but it is not possible to avoid
         // lateness if this time window is chosen. Schedule the daily rest
         // period.

         if red = 0 then
            scheduleActivity (i, duration, rest)
         else
            scheduleActivity (i, duration, redrest)
         end if

         Update driver status

      end if // if dailyTimeAfterService − hadDailyRest · waitingTime ...
    end if // if ptwr + duration − ...
  else

    // The time suffices to take a regular daily rest period. Note that the daily
    // rest period may end after the start of the time window.

    scheduleActivity (i, duration, rest)
    Update driver status

    break

  end if// if ptwr + duration > chosenTWEnd
else

  // An additional daily rest period is not necessary.
  // If lateness occurs, the current time window considered is already the last
  // one, as we started this loop with the "first reachable time window".

  break

end if // if dailyTimeAfterService > 780 + 120 (hpr + red)

lateness ← max (0, ptwr − chosenTWEnd)
```

until $lateness = 0 \vee z = nbTW_{i+1}$

// In the repeat-loop, z was raised by 1 one time too often. Therefore, subtract 1.

$z \leftarrow z - 1$

The method *reduceRestDurationLastRest*() in Algorithm 7 is used to reduce the the duration of the last daily rest period if this helps to reduce lateness. Waiting time may be compensated by extending the last daily rest period on the current arc (method *extendRestDurationLastRest* (< *extension in min.* >)). If there is waiting time, the method *extendRestDurationLastRest* (< *extension in min.* >) can be used to extend the last daily rest period on the current arc accordingly.

Algorithm 7 Schedule activities at stop $i + 1$

$lateness \leftarrow \max\left(0, ptwr - \overline{TW}^{end}_{i+1,z}\right)$

// If there is lateness, reduce the duration of daily rest periods.

if $lateness > 0$ **then**

 $posNoReductions = \min\left(getNoReducableRestPeriods(), 3 - noRed\right)$

 while $(ptwr > \overline{TW}^{end}_{i+1,z}) \wedge (posNoReductions > 0)$ **do**

  $reduceRestDurationLastRest()$
  $posNoReductions \leftarrow posNoReductions - 1$

 end while

else

 // ———————————————————————————
 // Plan activities after the arrival at the customer location.
 // ———————————————————————————

 // If there is waiting time, try to compensate it by resting activities.
 // If the last activity scheduled was a (reduced) daily rest period ignore this step.

 $noActivities \leftarrow getNoActivitiesArc()$
 if $noActivities < 1 \vee getActivityType\,(noActivities - 1) \neq "rest"$ **then**
  if $hpr = 1 \vee red = 1$ **then**
   $duration \leftarrow 540$
  else
   $duration \leftarrow 660$

end if

if $ptwr \leq \overline{TW}_{i+1,z}^{begin} - duration$ **then**
  if $red = 1$ **then**
    $scheduleActivity\,(i, duration, redrest)$
    Update driver status
  else
    $scheduleActivity\,(i, duration, rest)$
    Update driver status
  end if

else if $(ptwr \leq \overline{TW}_{i+1,z}^{begin} - 540) \wedge (noRed < 3)$ **then**
  $scheduleActivity\,(i, 540, redrest)$
  Update driver status

else if $(hpr = 0) \wedge (red = 0) \wedge (ptwr \leq \overline{TW}_{i+1,z}^{begin} - 180)$ **then**
  $scheduleActivity\,(i, 180, rest)$
  Update driver status

else if $(ptwr \leq \overline{TW}_{i+1,z}^{begin} - 45 + 15\ hpb) \wedge (udt > 0)$ **then**
  $scheduleActivity\,(i, 45 - 15\ hpb, break)$
  Update driver status

else if $(ptwr \leq \overline{TW}_{i+1,z}^{begin} - 15) \wedge (hpb = 0)$ **then**
  $scheduleActivity\,(i, 15, rest)$
  Update driver status

end if // if $ptwr \leq \overline{TW}_{i+1,z}^{begin} - duration$
end if // if $noActivities < 1 \vee getActivityType\,(noActivities - 1) \neq "rest"$
end if // if $lateness > 0$

// ————————————————————————————————
// Postprocessing: Compensate waiting time by extending the duration of the last
// daily rest period if possible.
// ————————————————————————————————
if $ptwr \leq \overline{TW}_{i+1,z}^{begin}$ **then**
  if $getDailyRestPosSize() > 0$ **then**
    $extendRestDurationLastRest\left(\overline{TW}_{i+1,z}^{begin} - ptwr\right)$

  else
    Save the driver status.
    Save the chosen time window.
    $scheduleActivity\left(i, \overline{TW}_{i+1,z}^{begin} - ptwr, wait\right)$

  end if // if $getDailyRestPosSize() > 0$

else
    Save the driver status.
    Save the chosen time window.
end if // if $ptwr \leq \overline{TW}_{i+1,z}^{begin}$

// ————————————————————————————————
// Plan loading or unloading at customer location.
// ————————————————————————————————
if $\bar{\Delta}_{i+1}^{service} > 0$ **then**
    $scheduleActivity\left(i, \bar{\Delta}_{i+1}^{service}, work\right)$
end if // if $\bar{\Delta}_{i+1}^{service} > 0$

C Examples of schedules

This appendix shows three examples of schedules. The aim is to give an idea of how the different planning techniques behave. Base instance 3 (see Figure 31) with 3 time windows and a time window size of 30 minutes (see Table 7) was chosen for the analysis. Tables 8 and 10 show the output of the transformation algorithm, Table 8 for the model without optional rules, and 10 for the model with optional rules. Table 9 depicts the result when using the myopic algorithm as a planning technique.

start	Mon 07:47						
target location			Rastatt (DE)	Kirkel (DE)	Madrid (ES)	Duenas (ES)	Wolfsburg (DE)
stops		[0]	[1]	[2]	[3]	[4]	[5]
dur. loading/unloading (h)			2:00	2:00	2:00	2:00	0:00
time windows	start		Mon 06:30	Mon 05:30	Wed 05:30	Thu 03:30	Mon 00:00
	end		Mon 07:00	Mon 06:00	Wed 06:00	Thu 04:00	Sun 23:59
	start		Mon 09:00	Mon 08:00	Wed 08:00	Thu 06:00	
	end		Mon 09:30	Mon 08:30	Wed 08:30	Thu 06:30	
	start		Mon 11:30	Mon 10:30	Wed 10:30	Thu 08:30	
	end		Mon 12:00	Mon 11:00	Wed 11:00	Thu 09:00	

Table 7: Time windows

It is interesting to see the advantages when optimizing over all arcs. Independent on the planning technique, the driver starts his work week at the first customer (no driving duration between start vertex 0 and customer vertex 1) at 7:47 on Monday morning. While the models decide to take the first time window and thus accept a lateness of 47 minutes, the myopic algorithm decides to avoid lateness at the first customer and chooses the second time window compensating part of the waiting time by a first partial break. The effect can be seen when having a look at the lateness at the second customer. While the myopic algorithm cannot avoid a lateness of 2:16 hours, the models manage to 'reduce' lateness to 1:03 hours. Over the first two customers, lateness can thus be reduced by 26 minutes.

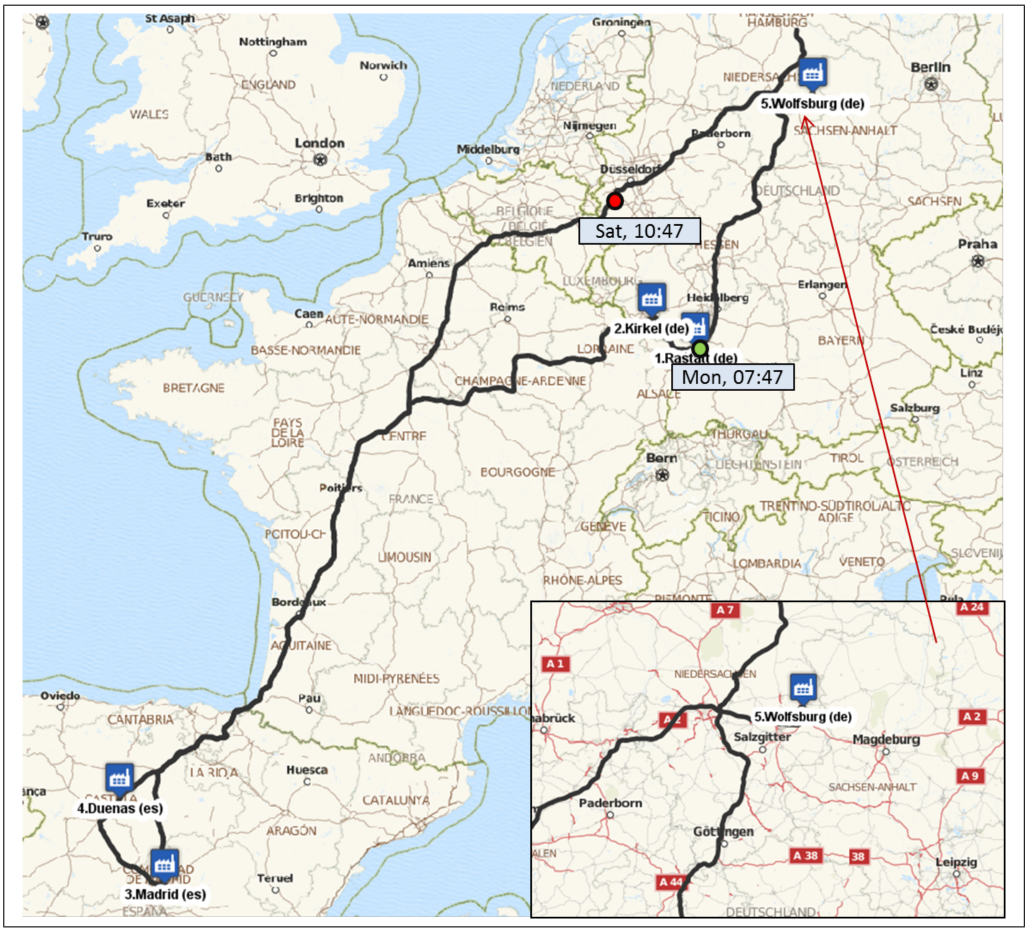

Figure 31: Route of base instance 3

The choice of time windows varies between the three schedules. The use of reduced daily rest periods allows the driver to be on time at stop three, no matter if the model with optional rules or the myopic algorithm is chosen. On the last arc the MILP model with consideration of the optional rules has the advantage to 'know' that there are no remaining requests in the considered week. All possible driving time extensions and reduced daily rest periods are used such that the driver is able to finish his tour significantly earlier than in the schedule that was created by the myopic algorithm. In turn, the myopic algorithm achieves a much earlier completion time than the MILP model without consideration of the optional rules.

	day	type	from	until	duration
	Mon	(un)load	07:47	09:47	02:00
stop: 1 chosen time window: start: calculated arrival: lateness:	 Mon Mon	 06:30 07:47 00:47	 end:	 Mon	 07:00
	Mon Mon	drive (un)load	09:47 12:03	12:03 14:03	02:16 02:00
stop: 2 chosen time window: start: calculated arrival: lateness:	 Mon Mon	 10:30 12:03 01:03	 end:	 Mon	 11:00
	Mon Mon Mon Mon Tue Tue Tue Tue Wed Wed Wed Wed	drive break drive rest drive break drive rest drive break drive (un)load	14:03 16:17 17:02 20:47 07:47 12:17 13:02 17:32 04:32 09:02 09:47 12:07	16:17 17:02 20:47 07:47 12:17 13:02 17:32 04:32 09:02 09:47 12:07 14:07	02:14 00:45 03:45 11:00 04:30 00:45 04:30 11:00 04:30 00:45 02:20 02:00
stop: 3 chosen time window: start: calculated arrival: lateness:	 Wed Wed	 10:30 12:07 01:07	 end:	 Wed	 11:00
	Wed Wed Thu Thu	drive rest drive (un)load	14:07 16:17 04:55 06:00	16:17 04:55 06:00 08:00	02:10 12:38 01:05 02:00
stop: 4 chosen time window: start: calculated arrival: lateness:	 Thu Thu	 06:00 06:00 00:00	 end:	 Thu	 06:30
	Thu Thu Thu Thu Fri Fri Fri Fri Sat	drive break drive rest drive break drive rest drive	08:00 11:25 12:10 16:40 03:40 08:10 08:55 13:25 00:25	11:25 12:10 16:40 03:40 08:10 08:55 13:25 00:25 03:07	03:25 00:45 04:30 11:00 04:30 00:45 04:30 11:00 02:42
stop:5 chosen time window: start: calculated arrival: lateness:	 Mon Sat	 00:00 03:07 00:00	 end:	 Sun	 23:59

Table 8: Optimal schedule identified by the MILP model without optional rules

day	type	from	until	duration	
	Mon	break	07:47	08:02	00:15
	Mon	wait	08:02	09:00	00:58
	Mon	(un)load	09:00	11:00	02:00
stop: 1					
chosen time window:					
start:	Mon	09:00	end:	Mon	09:30
calculated arrival:	Mon	08:02			
lateness:		00:00			
	Mon	drive	11:00	13:16	02:16
	Mon	(un)load	13:16	15:16	02:00
stop: 2					
chosen time window:					
start:	Mon	10:30	end:	Mon	11:00
calculated arrival:	Mon	13:16			
lateness:		02:16			
	Mon	drive	15:16	17:30	02:14
	Mon	break	17:30	18:00	00:30
	Mon	drive	18:00	20:47	02:47
	Mon	rest	20:47	05:47	09:00
	Tue	drive	05:47	10:17	04:30
	Tue	break	10:17	11:02	00:45
	Tue	drive	11:02	15:32	04:30
	Tue	rest	15:32	01:57	10:25
	Wed	drive	01:57	06:27	04:30
	Wed	break	06:27	07:12	00:45
	Wed	drive	07:12	10:30	03:18
	Wed	(un)load	10:30	12:30	02:00
stop: 3					
chosen time window:					
start:	Wed	10:30	end:	Wed	11:00
calculated arrival:	Wed	10:30			
lateness:		00:00			
	Wed	drive	12:30	13:42	01:12
	Wed	rest	13:42	00:42	11:00
	Thu	drive	00:42	02:45	02:03
	Thu	break	02:45	03:30	00:45
	Thu	(un)load	03:30	05:30	02:00
stop: 4					
chosen time window:					
start:	Thu	03:30	end:	Thu	04:00
calculated arrival:	Thu	03:30			
lateness:		00:00			
	Thu	drive	05:30	10:00	04:30
	Thu	break	10:00	10:45	00:45
	Thu	drive	10:45	13:12	02:27
	Thu	rest	13:12	00:12	11:00
	Fri	drive	00:12	04:42	04:30
	Fri	break	04:42	05:27	00:45
	Fri	drive	05:27	09:57	04:30
	Fri	rest	09:57	20:57	11:00
	Fri	drive	20:57	00:37	03:40
stop: 5					
chosen time window:					
start:	Mon	00:00	end:	Sun	23:59
calculated arrival:	Fri	20:57			
lateness:		00:00			

Table 9: Schedule created with the myopic algorithm

date	day	from	until	duration	
	Mon	(un)load	07:47	09:47	02:00
stop: 1					
chosen timekwindow:					
start:	Mon	06:30	end:	Mon	07:00
calculated arrival:	Mon	07:47			
lateness:		00:47			
	Mon	drive	09:47	12:03	02:16
	Mon	(un)load	12:03	14:03	02:00
stop: 2					
chosen time window:					
start:	Mon	10:30	end:	Mon	11:00
calculated arrival:	Mon	12:03			
lateness:		01:03			
	Mon	drive	14:03	16:17	02:14
	Mon	break	16:17	17:02	00:45
	Mon	drive	17:02	21:32	04:30
	Mon	rest	21:32	06:32	09:00
	Tue	drive	06:32	11:02	04:30
	Tue	break	11:02	11:47	00:45
	Tue	drive	11:47	16:17	04:30
	Tue	rest	16:17	01:17	09:00
	Wed	drive	01:17	05:47	04:30
	Wed	break	05:47	06:32	00:45
	Wed	drive	06:32	08:07	01:35
	Wed	(un)load	08:07	10:07	02:00
stop: 3					
chosen time window:					
start:	Wed	08:00	end:	Wed	08:30
calculated arrival:	Wed	08:07			
lateness:		00:00			
	Wed	drive	10:07	13:02	02:55
	Wed	rest	13:02	02:55	13:53
	Thu	drive	02:55	03:15	00:20
	Thu	break	03:15	03:30	00:15
	Thu	(un)load	03:30	05:30	02:00
stop: 4					
chosen time window:					
start:	Thu	03:30	end:	Thu	04:00
calculated arrival:	Thu	03:30			
lateness:		00:00			
	Thu	drive	05:30	09:40	04:10
	Thu	break	09:40	10:10	00:30
	Thu	drive	10:10	14:40	04:30
	Thu	break	14:40	15:25	00:45
	Thu	drive	15:25	16:25	01:00
	Thu	rest	16:25	01:25	09:00
	Fri	drive	01:25	05:55	04:30
	Fri	break	05:55	06:40	00:45
	Fri	drive	06:40	11:10	04:30
	Fri	break	11:10	11:55	00:45
	Fri	drive	11:55	12:52	00:57
stop: 5					
chosen time window:					
start:	Mon	00:00	end:	Sun	23:59
calculated arrival:	Fri	12:52			
lateness:		00:00			

Table 10: Optimal schedule identified by the MILP model with optional rules

References

Bartodziej, P., Derigs, U., Malcherek, D., and Vogel, U. (2009). Models and algorithms for solving combined vehicle and crew scheduling problems with rest constraints: an application to road feeder service planning in air cargo transportation. *OR Spectrum*, 31(2):405–429.

Council of the European Union. (1985). Council Regulation (EEC) No 3821/85 of 20 December 1985 on recording equipment in road transport. Official Journal of the European Communities.

Derigs, U., Kurowsky, R., and Vogel, U. (2011). Solving a real-world vehicle routing problem with multiple use of tractors and trailers and EU-regulations for drivers arising in air cargo road feeder services. *European Journal of Operational Research*, 213(1):309 – 319.

Deutscher Bundestag. (1994). Arbeitszeitgesetz vom 6. Juni 1994 (BGBl. I S. 1170, 1171), das zuletzt durch Artikel 3 Absatz 6 des Gesetzes vom 20. April 2013 (BGBl. I S. 868) geändert worden ist.

Drexl, M. and Prescott-Gagnon, E. (2010). Labelling algorithms for the elementary shortest path problem with resource constraints considering EU drivers' rules. *Logistics Research*, 2:79–96.

European Commission. (2009). Commission Directive 2009/4/EC of 23 January 2009 counter measures to prevent and detect manipulation of records of tachographs, amending Directive 2006/22/EC of the European Parliament and of the Council on minimum conditions for the implementation of Council Regulations (EEC) Nos 3820/85 and 3821/85 concerning social legislation relating to road transport activities and repealing Council Directive 88/599/EEC (Text with EEA relevance). Official Journal of the European Union.

European Commission. (2014). Report from the Commission to the European Parliament and the Council on the state of the union road transport market. Technical report, European Commission.

European Commission. (2016). Non EU-countries - Transport. URL `http://ec.europa.eu/transport/modes/road/non-eu-countries_en.htm`. Accessed on March 19, 2016.

European Parliament and Council of the European Union. (2002). Directive 2002/15/EC on the organisation of the working time of persons performing mobile road transport activities. Official Journal of the European Union.

European Parliament and Council of the European Union. (2006)a. Directive 2006/22/EC on minimum conditions for the implementation of Council Regulations (EEC) No 3820/85 and (EEC) No 3821/85 concerning social legislation relating to road transport activities and repealing Council Directive 88/599/EEC. Official Journal of the European Union.

European Parliament and Council of the European Union. (2006)b. Regulation (EC) No 561/2006 on the harmonisation of certain social legislation relating to road transport and amending Council Regulations (EEC) No 3821/85 and (EC) No 2135/98 and repealing Council Regulation (EEC) No 3820/85. Official Journal of the European Union.

European Parliament and Council of the European Union. (2014). Regulation (EU) No 165/2014 of the European Parliament and of the Council of 4 February 2014 on tachographs in road transport, repealing Council Regulation (EEC) No 3821/85 on recording equipment in road transport and amending Regulation (EC) No 561/2006 of the European Parliament and of the Council on the harmonisation of certain social legislation relating to road transport Text with EEA relevance. Official Journal of the European Union.

European Parliament, Council of the European Union. (2009). Regulation (EC) No 1073/2009 of the European Parliament and of the Council of 21 October 2009 on common rules for access to the international market for coach and bus services, and amending Regulation (EC) No 561/2006 (Text with EEA relevance). Official Journal of the European Union.

European Road Safety Observatory. (2015). Annual accident report 2015. URL `http://ec.europa.eu/transport/road_safety/specialist/statistics/index_en.htm`. Accessed on March 06, 2016.

Goel, A. (2009). Vehicle scheduling and routing with drivers' working hours. *Transportation Science*, 43(1):17–26.

Goel, A. (2010). Truck driver scheduling in the European Union. *Transportation Science*, 44(4):429–441.

Goel, A. (2012). The minimum duration truck driver scheduling problem. *EURO Journal on Transportation and Logistics*, 1(4):285–306.

Goel, A. and Gruhn, V. (2006). Drivers' working hours in vehicle routing and scheduling. In *Proceedings of the 9th IEEE International Conference on Intelligent Transportation Systems*, pages 1280–1285.

Goel, A. and Vidal, T. (2014). Hours of service regulations in road freight transport: An optimization-based international assessment. *Transportation Science*, 48(3):391–412.

Kok, A. L., Meyer, C. M., Kopfer, H., and Schutten, J. M. J. (2010). A dynamic programming heuristic for the vehicle routing problem with time windows and European Community social legislation. *Transportation Science*, 44(4):442–454.

Kok, A. L., Hans, E. W., Schutten, J. M. J., and Zijm, W. H. M. (2011)a. A dynamic programming heuristic for vehicle routing with time-dependent travel times and required breaks. *Flexible Services and Manufacturing Journal*, 22(1):83–108.

Kok, A. L. (2010). *Congestion avoidance and break scheduling within vehicle routing*. PhD thesis, Beta Research School for Operations Management and Logistics, University of Twente, Enschede.

Kok, A., Hans, E., and Schutten, J. (2011)b. Optimizing departure times in vehicle routes. *European Journal of Operational Research*, 210(3):579 – 587.

Kopfer, H. and Meyer, C. M. (2010). Ein Optimierungsmodell für die wöchentliche Tourenplanung unter Einbeziehung der EU-Sozialvorschriften. *Zeitschrift für Betriebswirtschaft*, 80(7):755–775.

Kopfer, H., Meyer, C. M., and Wagenknecht, A. (2007). Die EU-Sozialvorschriften und ihr Einfluss auf die Tourenplanung. *Logistik Management*, 9(2):32–47.

Kopfer, H. and Meyer, C. M. (2009). A model for the traveling salesman problem including the EC regulations on driving hours. In Tuma, A., Fleischmann, B., Borgwardt, K., and Klein, R., editors, *Operations Research Proceedings 2009*, pages 289–294. Springer.

Meyer, C. M. (2011). *Vehicle Routing under Consideration of Driving and Working Hours: A Distributed Decision Making Perspective*. Produktion und Logistik. Gabler Research. Universität Bremen, Dissertation.

Meyer, C. and Kopfer, H. (2008). Restrictions for the operational transportation planning by regulations on drivers' working hours. In Bortfeldt, A., Homberger, J., Kopfer, H., Pankratz, G., and Strangmeier, R., editors, *Intelligent Decision Support*, pages 177–186. Gabler.

Prescott-Gagnon, E., Desaulniers, G., Drexl, M., and Rousseau, L.-M. (2010). European driver rules in vehicle routing with time windows. *Transportation Science*, 44(4):455–473.

SafetyNet. (2009). Fatigue. URL `http://ec.europa.eu/transport/road_safety/specialist/knowledge/fatique/index_en.htm`. Accessed on March 06, 2016.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265.

United Nations Economic Commission for Europe. (2006). Harmonization of requirements concerning international road transport and facilitation of its operation. Consolidated text of the European Agreement concerning the work of crews of vehicles engaged in international road transport (AETR). ECE/TRANS/ SC.1/2006/2.

Williams, H. P. (2013). *Model Building in Mathematical Programming, 5th Edition*. Wiley, 5th edition.

Zäpfel, G. and Bögl, M. (2008). Multi-period vehicle routing and crew scheduling with outsourcing options. *International Journal of Production Economics*, 113(2):980–996.

Veröffentlichte Berichte der Fakultät für Wirtschaftswissenschaften

Die PDF-Dateien der folgenden Berichte sind verfügbar unter:

Published reports of the Saarland Business School

The PDF files of the following reports are available under:

http://www.htwsaar.de/wiwi

1 I. Correia, T. Melo, F. Saldanha da Gama

Comparing classical performance measures for a multi-period, two-echelon supply chain network design problem with sizing decisions

Keywords: supply chain network design, facility location, capacity acquisition, profit maximization, cost minimization

(43 pages, 2012)

2 T. Melo

A note on challenges and opportunities for Operations Research in hospital logistics

Keywords: Hospital logistics, Operations Research, application areas

(13 pages, 2012)

3 S. Hütter, A. Steinhaus

Forschung an Fachhochschulen – Treiber für Innovation im Mittelstand: Ergebnisse der Qbing-Trendumfrage 2013

Keywords: Innovation, Umfrage, Trendbarometer, Logistik-Konzepte, Logistik-Technologien, Mittelstand, KMU

(5 pages, 2012)

4 A. Steinhaus, S. Hütter

Leitfaden zur Implementierung von RFID in kleinen und mittelständischen Unternehmen

Keywords: RFID, KMU, schlanke Prozesse, Prozessoptimierung, Produktion, Forschungsgruppe Qbing

(49 pages, 2013)

5 S.A. Alumur, B.Y. Kara, M.T. Melo

Location and Logistics

Keywords: forward logistics network design, reverse logistics network design, models, applications

(26 pages, 2013)

6 S. Hütter, A. Steinhaus

Forschung an Fachhochschulen – Treiber für Innovation im Mittelstand: Ergebnisse der Qbing-Trendumfrage 2013

Keywords: Innovation, Umfrage, Trendbarometer, Logistik-Konzepte, Logistik-Technologien, Mittelstand, KMU

(6 pages, 2014)

7 M.J. Cortinhal, M.J. Lopes, M.T. Melo

Redesigning a three-echelon logistics network over multiple time periods with transportation mode selection and outsourcing opportunities

Keywords: logistics network design/re-design, multiple periods, transportation mode selection, product outsourcing, mixed-integer linear programming

(49 pages, 2014)

8 T. Bousonville, C. Ebert, J. Rath

A comparison of reward systems for truck drivers based on telematics data and driving behavior assessments

Keywords: Telematics, driving behavior, incentives, award systems

(9 pages, 2015)

9 I. Correia, T. Melo

Multi-period capacitated facility location under delayed demand satisfaction

Keywords: Location, multi-period, capacity choice, delivery lateness, MILP models

(35 pages, 2015)

10 C.L. Martins, M.T. Melo, M.V. Pato

Redesigning a food bank supply chain network, Part I: Background and mathematical formulation

Keywords: Supply chain, sustainability, tri-objective problem, MILP model

(30 pages, 2016)

11 I. Correia, T. Melo

A computational comparison of formulations for a multi-period facility location problem with modular capacity adjustments and flexible demand fulfillment

Keywords: Facility location, multi-period, capacity expansion and contraction, delivery lateness, mixed-integer linear models

(42 pages, 2016)

12 A. Bernhardt, T. Melo, T. Bousonville, H. Kopfer

Scheduling of driver activities with multiple soft time windows considering European regulations on rest periods and breaks

Keywords: road transportation, driver scheduling, rest periods, breaks, driving hours, Regulation (EC) No 561/2006, mixed integer linear programming models

(137 pages, 2016)

htw saar

Hochschule für Technik und Wirtschaft des Saarlandes
University of Applied Sciences

Hochschule für Technik und Wirtschaft des Saarlandes

Die Hochschule für Technik und Wirtschaft des Saarlandes (htw saar) wurde im Jahre 1971 als saarländische Fachhochschule gegründet. Insgesamt studieren rund 5000 Studentinnen und Studenten in 38 verschiedenen Studiengängen an der htw saar, aufgeteilt auf vier Fakultäten.

In den vergangenen zwanzig Jahren hat die Logistik immens an Bedeutung gewonnen. Die htw saar hat dieser Entwicklung frühzeitig Rechnung getragen und einschlägige Studienprogramme sowie signifikante Forschungs- und Technologietransferaktivitäten entwickelt. Die Veröffentlichung der Schriftenreihe Logistik soll die Ergebnisse aus Forschung und Projektpraxis der Öffentlichkeit zugänglich machen.

Weitere Informationen finden Sie unter **http://logistik.htwsaar.de**

ISCOM
INSTITUT FÜR SUPPLY CHAIN
UND OPERATIONS MANAGEMENT

Institut für Supply Chain und Operations Management

Das Institut für Supply Chain und Operations Management (ISCOM) der htw saar ist auf die Anwendung quantitativer Methoden in der Logistik und deren Implementierung in IT-Systemen spezialisiert. Neben öffentlich geförderten Forschungsprojekten zu innovativen Themen arbeitet ISCOM eng mit Projektpartnern aus der Wirtschaft zusammen, wodurch der Wissens- und Technologietransfer in die Praxis gewährleistet wird. Zu den Arbeitsgebieten zählen unter anderem Distributions- und Transportplanung, Supply Chain Design, Bestandsmanagement in Supply Chains, Materialflussanalyse und -gestaltung sowie Revenue Management.

Weitere Informationen finden Sie unter **http://iscom.htwsaar.de**

Qbing
Produktion | Logistik | Technologie

Forschungsgruppe Qbing

Qbing ist eine Forschungsgruppe an der Hochschule für Technik und Wirtschaft des Saarlandes, die spezialisiert ist auf interdisziplinäre Projekte in den Bereichen Produktion, Logistik und Technologie. Ein Team aus derzeit acht Ingenieuren und Logistikexperten arbeitet unter der wissenschaftlichen Leitung von Prof. Dr. Steffen Hütter sowohl in öffentlich geförderten Projekten als auch zusammen mit Industriepartnern an aktuellen Fragestellungen zur Optimierung von logistischen Prozessabläufen in Handel und Industrie unter Einbeziehung modernster Sensortechnologie und Telemetrie. Qbing hat auch und gerade auf dem Gebiet der angewandten Forschung Erfahrung in der Zusammenarbeit mit kleinen und mittelständischen Unternehmen.

Weitere Informationen finden Sie unter **http://www.qbing.de**

ISSN 2193-7761