

Drexl, Michael; Rieck, Julia; Sigl, Thomas; Press, Bettina

Article

Simultaneous Vehicle and Crew Routing and Scheduling for Partial- and Full-Load Long-Distance Road Transport

BuR - Business Research

Provided in Cooperation with:

VHB - Verband der Hochschullehrer für Betriebswirtschaft, German Academic Association of Business Research

Suggested Citation: Drexl, Michael; Rieck, Julia; Sigl, Thomas; Press, Bettina (2013) : Simultaneous Vehicle and Crew Routing and Scheduling for Partial- and Full-Load Long-Distance Road Transport, BuR - Business Research, ISSN 1866-8658, VHB - Verband der Hochschullehrer für Betriebswirtschaft, German Academic Association of Business Research, Göttingen, Vol. 6, Iss. 2, pp. 242-264,
<https://doi.org/10.1007/BF03342751>

This Version is available at:

<https://hdl.handle.net/10419/103728>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



Simultaneous Vehicle and Crew Routing and Scheduling for Partial- and Full-Load Long-Distance Road Transport

Michael Drexl, Researcher at Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz and Fraunhofer Centre for Applied Research on Supply Chain Services SCS, Nuremberg, Germany, E-mail: drexl@uni-mainz.de

Julia Rieck, Operations Research Group, Institute of Management and Economics, Clausthal University of Technology, Germany, E-mail: julia.rieck@tu-clausthal.de

Thomas Sigl, Fraunhofer Centre for Applied Research on Supply Chain Services SCS, Nuremberg, Germany, E-mail: thomas.sigl@web.de

Bettina Press, Fraunhofer Centre for Applied Research on Supply Chain Services SCS, Nuremberg, Germany, E-mail: bettina.press@scs.fraunhofer.de

Abstract

This paper studies a simultaneous vehicle and crew routing and scheduling problem arising in long-distance road transport in Europe: Pickup-and-delivery requests have to be fulfilled over a multi-period planning horizon by a heterogeneous fleet of trucks and drivers. Typically, in the vehicle routing literature, a fixed assignment of a driver to a truck is assumed. In our approach, we abandon this assumption and allow truck/driver changes at geographically dispersed relay stations. This offers greater planning flexibility and allows a better utilization of trucks, but also creates intricate interdependencies between trucks and drivers and requires the synchronization of their routes. A solution heuristic based on a two-stage decomposition of the problem is developed, taking into account European Union social legislation for drivers, and computational experiments using real-world data provided by a major German forwarder are presented and analyzed. The obtained results suggest that for the vehicle and driver cost structure prevalent in Western Europe and for transport requests that are not systematically acquired to complement one another, no cost savings are possible through simultaneous vehicle and crew routing and scheduling, although no formal proof of this fact is possible.

JEL classification: C65, M29

Keywords: vehicle routing, crew scheduling, synchronization, full truckload transportation

Manuscript received October 2, 2012, accepted by Karl Inderfurth (Operations and Information Systems) June 12, 2013.

1 Introduction

Vehicle routing problems (VRPs) are among the most studied mathematical optimization problems; see, for example, the monographs [Toth and Vigo \(2002\)](#) and [Golden, Raghavan, and Wasil \(2008\)](#). This interest results from the importance of vehicle routing in everyday logistics practice as well as the intellectual challenge of finding feasible or optimal VRP solutions; VRPs belong to the class of NP-hard optimization problems ([Lenstra and Rinnooy Kan 1981](#)). The widespread practical use of mathematical models and algorithms for the solution of VRPs, as reflected by the large number of vendors of commercial vehicle routing software,

shows that the efforts of science in this field have had a significant impact on business operations. One application area, however, where there is a large gap between practical requirements and pertinent scientific work is simultaneous vehicle and crew routing and scheduling. This term denotes the situation where the required transports *have no given timetable/no fixed schedule*, and where a driver-vehicle combination is no longer considered an inseparable unit. Thus, routes to perform the required transports have to be planned for both vehicles and drivers. This is an important issue in practice, because drivers regularly need breaks and rests and have to obey social legislation or

trade union rules regarding driving, break, and rest times, whereas vehicles can be used twenty-four hours a day. Thus, the planning of separate vehicle and driver routes offers greater planning flexibility and allows a better temporal utilization of vehicles. In the present paper, we consider the European Union social legislation for drivers.

The large majority of papers on vehicle routing, nonetheless, does not distinguish between a vehicle and its driver. In their monograph on VRPs, for example, [Toth and Vigo \(2002\)](#) state that throughout, “the constraints imposed on drivers are imbedded in those associated with the corresponding vehicles”. Hence, the contribution of the present paper is to study how separate, but synchronized, routes for drivers and vehicles can be computed for transports without a given timetable. The remainder of the paper is structured as follows. In the next section, we give a description of the proposed simultaneous vehicle and crew routing and scheduling problem (SVCRRSP). Then, in Section 3, the relevant literature is briefly surveyed. Section 4 points out the distinguishing features of the present problem. Section 5 gives an overview of how the European social legislation on driver driving and working time is taken into account in this paper. In Section 6, a two-stage heuristic algorithm for solving real-world instances of the SVCRRSP is described. Computational experiments with the algorithm are reported in Section 7. The paper closes with a summary and a research outlook in Section 8.

2 Problem description

In order to narrow the above-mentioned gap, we studied the following simultaneous vehicle and crew routing and scheduling problem. Partial and complete loads (henceforth referred to as requests) have to be transported by truck from pickup to delivery locations. There is a time window for the pickup and for the delivery of each request. For the transport of the requests, a fleet of trucks and a set of drivers are available. Each driver and each truck has a fixed home depot, where he/it is positioned at the beginning of the planning horizon and to where he/it must return at the end of the planning horizon. The planning horizon spans one working week, that is, six days from Monday to Saturday. Thus, a driver/truck route starts and ends at an assigned depot and may cover several days.

The truck fleet is heterogeneous; there are several types of trucks that may differ with respect to costs, capacity, and ability to perform certain requests. The drivers are homogeneous in the sense that each driver is able to drive each truck and that all drivers receive the same wage. In addition to the home depots and the pickup and delivery locations, there are so-called intermediate depots or *relay stations*. It is possible that a location acts as both a home depot and a relay station. At relay stations, and only there, drivers can change trucks. Hence, there is a free assignment of trucks to drivers and vice versa, that is, a driver may change trucks arbitrarily often within the planning horizon, respectively, a truck may be driven by arbitrarily many drivers within the planning horizon. The fact that there are multiple depots, where drivers and trucks may be stationed, and relay stations, where a change of truck or driver is possible, makes the synchronization of drivers and trucks a non-trivial issue. In addition to the requirements already stated, the drivers have to fulfill the European Union social legislation on driver driving and working hours on single-manned vehicles. It is assumed that a driver can take a break or rest anywhere en route. However, a driver must visit a relay station and take a daily rest there, or he must finish his route at his home depot, upon expiry of a given fixed period after leaving his home depot or ending a daily rest at a previous relay station. Note that the relay stations have no capacity restrictions, that is, the overall number of driver-vehicle changes at a relay station as well as the number of simultaneous changes is unlimited. Furthermore, we consider the option of driver shuttle transports, where small shuttle vans are used to transport truck drivers between relay stations. In doing so, a truck driver is able to leave a truck at a relay station and subsequently drive a truck starting from a different relay station. The shuttle transports are not counted as driving or working time for the truck drivers. The possibility of shuttle transports increases the flexibility for matching trucks and drivers, but of course shuttle transports also induce costs. A typical objective for the SVCRRSP is to minimize the overall operating costs, which consist of truck, driver, and shuttle costs.

To fulfill a request, two different objects (a driver and a truck) must be synchronized in space and time, since both are non-autonomous objects unable to move in space on their own. From these

synchronization requirements, the central additional difficulty of the SVCRSP compared to classical VRPs results, the so-called *interdependence problem*:

The interdependence problem refers to the fact that a change in one route may have effects on the feasibility of other routes.

This is in marked contrast to classical VRPs without synchronization, where any operation modifying a subset of all existing routes leaves the remaining routes unaffected. Indeed, most VRP solution algorithms, notably local search procedures and approaches based on column generation, implicitly assume independence of routes. Consequently, standard VRP solution procedures known from the literature are not directly applicable to solving SVCRSPs. Hence, it is necessary to use mechanisms for dealing with the synchronization requirements and the resulting interdependence problem.

The SVCRSP is a special case of the broader class of *vehicle routing problems with multiple synchronization constraints (VRPMSs)*. These are VRPs where the routes of different ‘vehicles’ are interdependent. For a classification of VRPMSs, the reader is referred to [Drexl \(2012\)](#). This paper also provides a comprehensive overview of the existing literature on synchronization in vehicle routing. In the following section, the most important papers influencing our research are briefly discussed.

3 Literature review

The works to be mentioned here may be partitioned into two classes: First, papers considering cases where motor vehicles and drivers are synchronized at a central depot and where a change of driver/vehicle is possible only at the depot. Second, papers where different types of elementary autonomous and/or non-autonomous objects are required to fulfill tasks, and where these objects may join and separate at several different locations. Compared to the first class, the locational flexibility of the second class adds an additional degree of freedom. Hence, problems in the second class turn out to be significantly harder to solve than those in the first class.

Synchronization at a central depot. [Laurent and Hao \(2007\)](#) consider the problem of simultaneously scheduling vehicles and drivers for a limousine rental company. The required transports are

pickup-and-delivery trips with given time windows. The authors use a two-stage solution approach that aims to find a feasible crew and vehicle schedule by assigning a driver-limousine pair to each trip. First, an initial feasible solution is constructed by means of a greedy heuristic similar to the well-known best-fit-decreasing strategy for the bin packing problem, using constraint programming techniques for domain reduction. Second, an improvement procedure based on local search embedded in a simulated annealing metaheuristic is performed.

[Prescott-Gagnon, Desaulniers, and Rousseau \(2010\)](#) study the problem of planning oil deliveries to customers by truck. To solve the problem, the authors develop three metaheuristics, a tabu search (TS) algorithm, a large neighborhood search (LNS) heuristic based on this TS algorithm and another LNS heuristic based on a column generation (CG) heuristic that uses the TS algorithm to generate columns. In addition, a greedy construction heuristic is considered that sequentially builds up routes for driver-truck pairs by inserting the temporally closest customer. In the destruction phase of the LNS algorithm, different heuristics similar to those presented in [Ropke and Pisinger \(2006\)](#) are used to determine the vertices to be removed from the routes of the current solution. The reconstruction phase applies either TS or CG. Among other move types, the TS procedure employs a driver switch move that tries to switch a pair of drivers, that is, have one driver drive the other driver’s route and vice versa.

[Xiang, Chu, and Chen \(2006\)](#) describe a static dial-a-ride problem that involves the scheduling of heterogeneous vehicles and a group of drivers with different qualifications. The solution procedure is composed of a construction phase to obtain an initial solution, an improvement phase, and an intensification phase to fine-tune the solution. The important aspect of the procedure is that, initially, ‘abstract’ routes with a fixed schedule are determined, and only in the last stage, concrete vehicles and drivers are assigned to the routes.

[Zäpfel and Bögl \(2008\)](#) consider an application of local letter mail distribution. Pickup routes and delivery routes (but no combined pickup-and-delivery routes) have to be planned within a planning horizon of one week. In pickup routes, outbound shipments are transported from local post offices to a letter mail distribution center. Con-

versely, in delivery routes, shipments are transported from the distribution center to post offices. Schedules are planned for both drivers and vehicles, taking into account European Union social legislation. The problem is solved heuristically, by decomposing it into a generalized VRP with time windows (GVRPTW) and a ‘personnel assignment problem’ (PAP). First, a feasible solution to the GVRPTW is computed via a modification of the I1 heuristic by [Solomon \(1987\)](#). Then, the PAP tries to find a feasible driver assignment for the GVRPTW solution. The assignment is achieved by creating a table with all feasible combinations of drivers and routes. Each table entry represents the costs resulting from the driver performing the route. A complete personal assignment is computed, using three different strategies, among them a greedy and a random procedure. After that, an improvement procedure embedded into a metaheuristic follows.

Synchronization en route. [Kim, Koo, and Park \(2010\)](#) study a combined vehicle routing and staff scheduling problem where a certain number of tasks has to be fulfilled in a fixed sequence at customers. Among the tasks, an end-to-start relationship is assumed. In order to fulfill the tasks, different teams of workers are available. Each team is qualified to perform one specific type of task. The teams cannot move by themselves; instead, a set of vehicles is used to transport the teams. There is no fixed assignment of a vehicle to a team, and each vehicle may carry at most one team at a time. The authors develop an astonishingly simple procedure, in which the vehicles, the teams, and the next tasks for each customer are stored in three lists, along with the relevant information on times and locations. In each iteration, a triplet (vehicle, team, task) is selected from the lists, using a best-fit criterion. Then, the lists are updated to reflect the situation resulting when the selected vehicle transports the selected team to the location of the selected task.

[Hollis, Forbes, and Douglas \(2006\)](#) describe a simultaneous vehicle and crew routing and scheduling application for urban letter mail distribution at Australia Post. The authors are the first to consider a problem with multiple depots, where vehicles and drivers may be stationed and interchanged. In order to solve the problem, a two-stage approach is used. In the first stage, the authors determine ‘abstract’ vehicle routes by solving a

pickup-and-delivery problem with time windows, multiple depots, a heterogeneous fleet of vehicles as well as several working time restrictions for drivers. A path-based mixed-integer programming (MIP) model is presented and solved by heuristic column generation. In the second stage, concrete vehicle and crew schedules are determined taking an integrated vehicle and crew *scheduling* approach. This is again done by solving an MIP with heuristic column generation. In the second-stage MIP, the tasks to be performed correspond to the vehicle routes computed in the first stage.

It is noteworthy that, for synchronization at a central depot as well as synchronization en route, although the objects to be synchronized in the papers described are vehicles and drivers, the concrete application contexts are almost all different (ranging from limousine rental to mail distribution). As regards solution approaches, and, in particular, the consideration of the interdependence problem, [Zäpfel and Bögl \(2008\)](#) perform, in principle, a decomposition of the problem by object type, exploiting the fact that drivers and vehicles may join and separate at one location only. On the other hand, [Laurent and Hao \(2007\)](#) and [Prescott-Gagnon, Desaulniers, and Rousseau \(2010\)](#) take the opposite way and compute routes for predetermined driver-vehicle pairs. [Xiang, Chu, and Chen \(2006\)](#) and [Hollis, Forbes, and Douglas \(2006\)](#) determine abstract routes first, and assign concrete vehicles and drivers afterwards. [Kim, Koo, and Park \(2010\)](#) use a heuristic in which a direct selection of vehicles, teams, and tasks is performed.

A recurring idea in most solution procedures described above is to decompose the problem into several stages and, in early stages, partially take aspects into account that are important to be able to obtain feasible solutions in later stages. Our solution approach, described in Section 6, also follows this principle.

As described in Section 2, the problem under consideration here is an SVCRSP in which European Union social legislation on drivers’ driving and working times (driver rules for short) are considered. As mentioned, the relevant rules are briefly described in Section 5; detailed information on these regulations is given in the Appendix. Procedures for considering them in VRP algorithms are described, for example, in [Goel \(2009\)](#), [Drexel and Prescott-Gagnon \(2010\)](#), [Goel \(2010\)](#), [Kok, Meyer, Kopfer, and Schutten \(2010\)](#), and [Prescott-](#)

Gagnon, Desaulniers, Drexl, and Rousseau (2010). Our SVCRSP algorithm uses the procedure described in the last reference. Driver rules are in force also in other countries or regions throughout the world. As we focus here on the situation in Europe, considering rules other than the European ones is beyond the scope of this paper.

A related and well studied field of application is integrated vehicle and crew *scheduling*, for example in the airline business or in the public transport sector. Pertinent surveys are given in Klabjan (2005), Caprara, Kroon, Monaci, Peeters, and Toth (2007), and Desaulniers and Hickman (2007). However, the models and solution approaches used for solving such problems are always based on the fundamental assumption of a *given timetable* for the requests/tasks. Since we assume here that the required transports do *not* have a given timetable, the SVCRSP requires different or at least modified solution methods.

4 Important features of the problem

The important and distinguishing features of the present problem compared to the ones presented in the references discussed above lie in the consideration of the following criteria:

- Practical and real-life assumptions:
 - Shipments/requests may originate and end anywhere, not only at truck or driver depots.
 - A route segment between two relay stations may be driven by any driver.
 - A truck need not be empty when a driver change is performed at a relay station.
 - Driver shuttle transports to and from relay stations are possible.
 - European Union social legislation on driver driving and working times is considered (see next section).
- Components of the solution approach:
 - The algorithm is based on a heuristic decomposition of the overall problem into two stages. However, compared to the highly sophisticated approaches taken in some of the above papers, our algorithm is rather simple and straightforward and primarily relies on an appropriate network representation of the problem.
 - Both stages can be solved with essentially the same algorithm.
 - The large neighborhood search heuristic we

use can very easily be replaced by any other metaheuristic.

5 Driver rules

The current EU legislation specifies a complex set of rules on driver driving and working times (for a detailed discussion see the Appendix). For our work, we consider the following rules:

- A break of at least 45 minutes is required
 - after 4.5 hours of driving, or
 - after 6 hours of work.
- A daily rest of at least 11 hours is required
 - after 9 hours of driving, or
 - after 9.6 hours of work, or
 - 13 hours of wall-clock time after the end of the last daily rest.
- A weekly rest of at least 24 hours must be taken
 - after 45 hours of driving, or
 - after 48 hours of working, or
 - 144 hours (6 days) of wall-clock time after the end of the last weekly rest.

‘Work’ encompasses driving as well as complementary activities such as loading or unloading, paperwork etc.

Driver schedules that comply with these rules may be considered legal. The EU rules provide additional degrees of freedom and allow splitting of breaks or rests, shortening daily rests, and extending daily or weekly driving and working times. We do not consider these options for two reasons: First, taking all these options into account would be too computationally costly for the large instances we attempt to solve. Second, not considering optional rules makes the resulting route plans and schedules more robust and thus more suitable for practice: In the case of unexpected events, such as detours, traffic jams, vehicle breakdowns etc., the possibility to shorten a rest or increase driving or working time may help to stick to an original plan and fulfill all requests in time.

Thus, the relevant resources concerning driver rules are the times since the end of the last weekly and daily rest, the weekly, daily and interval driving times, and the weekly, daily and interval working times. At this, the daily driving (working) time is the total accumulated driving (working) time between two daily rests and the interval driving (working) time is the total accumulated driving (working) time between two breaks. At the beginning of the planning horizon, each truck is located

at its home depot, and all drivers are considered rested, that is, the aforementioned time resources are all set to zero.

Note that the law makes no requirements with respect to the location where breaks or daily or weekly rests must be taken; these may be taken anywhere en route, such as on customer premises, at public parking places, or even at the roadside. All the references on driver rules cited in the literature review above assume that breaks or rests can be taken directly when scheduled and without incurring any detour to or from a suitable location. That is, it is essentially assumed that the driver simply stops at the roadside to take a break or rest if he is away from a depot or a relay station or a customer location. By allowing breaks or daily rests to be taken anywhere, specifying a maximum time period for two subsequent visits at a relay station (henceforth called *inter-visit time*), and requiring that a daily rest must be taken before leaving a relay station with a truck, we follow this established approach. (In the computational experiments, different values for the inter-visit time were chosen, see Section 7).

6 Solution approach

As mentioned, the basic idea of our solution algorithm is to take a two-stage approach. In the first stage, routes for trucks are determined, taking into account some driver rules. In the second stage, routes for drivers are computed, based on the truck routes from stage 1 and taking into account the remaining driver rules to ensure feasibility. We thus adopt some ideas from [Hollis, Forbes, and Douglas \(2006\)](#), but in our first stage, routes are computed for concrete trucks.

More precisely, the first stage consists in the solution of a pickup-and-delivery problem with time windows, relay stations, and additional constraints (PDPTWRS). The vehicles in the PDPTWRS correspond to the trucks of the underlying SVCRRSP. There are multiple depots, namely, the home depots of the trucks. Each truck performs at most one route, and the maximum route duration is set to the length of the planning horizon. For each truck, driver rules are considered in a manner described in detail below. At the end of the planning horizon, each truck must return to its home depot.

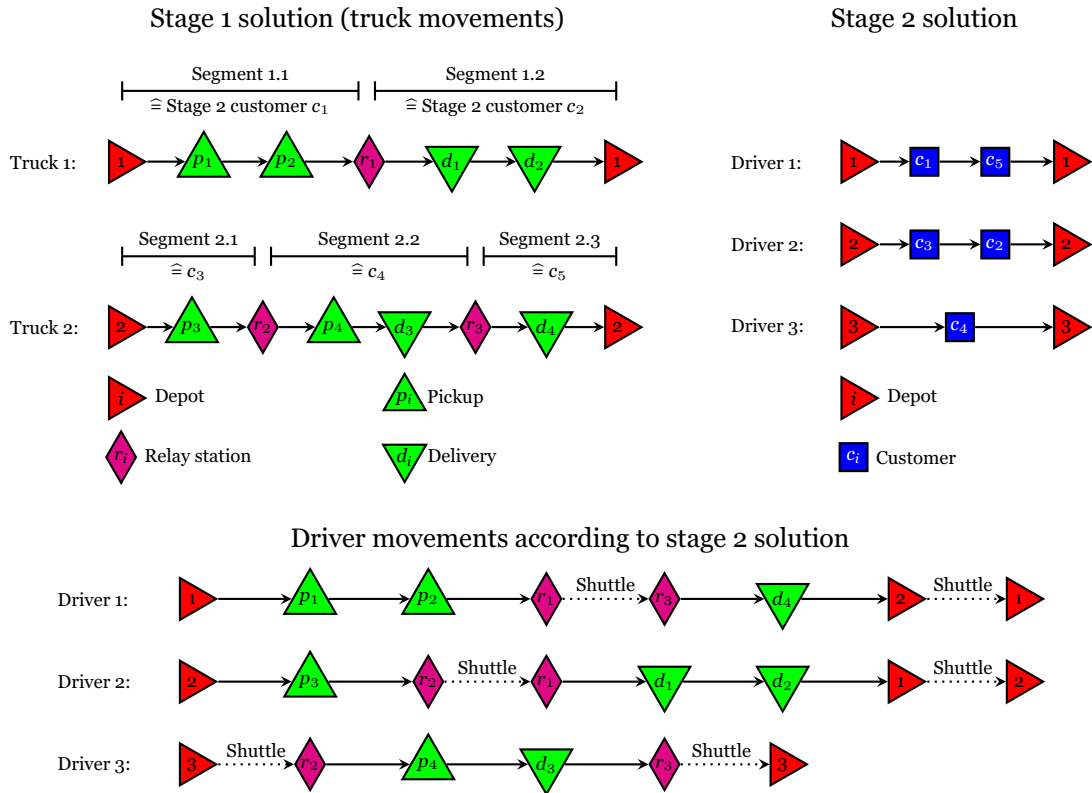
The second stage consists in the solution of a vehicle routing problem with time windows and

multiple depots (VRPTWMD). In this problem, the partial truck routes or segments starting and/or ending at a home depot or a relay station, as determined in the first stage, are the customers. The drivers are routed according to the solution of the second-stage problem, that is, the drivers act as the vehicles in the VRPTWMD. Each driver drives at most one route, and the maximum route duration is again set to the length of the planning horizon. Figure 1 shows the relationship between the two stages based on an example instance.

We consider an instance with three depots (1, 2, 3), three relay stations (r_1, r_2, r_3), and four pickup-and-delivery requests ($p_i \rightarrow d_i, i = 1, \dots, 4$). A possible solution of stage 1, depicted on the left side of the upper part of the figure, consists of two truck routes. The route of truck 1 starts at the first depot, picks up the two requests 1 and 2, visits relay station r_1 , delivers the two requests 1 and 2, and returns to the depot. Consequently, the route is comprised of two segments. The first one, segment 1.1, goes from depot 1 to relay station r_1 , and yields customer c_1 for stage 2 of the solution procedure. The second segment goes from relay station r_1 back to the depot, and yields customer c_2 for stage 2. Similarly, the route of truck 2 consists of three segments and, hence, yields three customers for stage 2. A possible solution of stage 2, depicted on the right side of the upper part of the figure, contains three driver routes. Driver 1 starts at depot 1, ‘visits’ the two ‘customers’ c_1 and c_5 , and returns to the depot. Driver 2 starts at depot 2, visits customers c_3 and c_2 , and returns. Finally, driver 3 starts at depot 3, visits customer c_4 , and returns. Each shape on the left-hand side corresponds to a physical location, and the depicted routes show the exact movements of the two trucks. On the right-hand side, only the triangles correspond to a physical location. The rectangles represent, in compact form, a movement in space. The lower part of the figure illustrates the exact movements of the drivers, that is, ‘extracts’ the route information hidden in the stage 2 customers. There, again, all shapes correspond to physical locations.

Note that the idea to model route segments as customers in a VRPTW was previously used in the already mentioned paper by [Hollis, Forbes, and Douglas \(2006\)](#) as well as in [Fügenschuh \(2006\)](#). In the latter paper, however, there is no simultaneous vehicle and driver routing and scheduling and thus no stage 1.

Figure 1: Relationship between stages 1 and 2



In the following, the main features of the problems in the two stages are summed up.

Stage 1 problem:

Pickup-and-delivery problem with time windows and relay stations (PDPTWRS):

- Pickup-and-delivery requests with given capacity requirements (weight, pallet places) and loading and unloading times
- Hard, single time windows at locations
- Vehicles stationed at different depots, and differing with respect to capacities and ability to perform requests
- Each vehicle performs at most one route starting and ending at specified depots (start and end depot may be different)
- Maximum route duration equal to the length of the planning horizon (6 days)
- Maximum time between two visits at a relay station
- Driver rules:
 - Break of 45 minutes after 4.5 hours of driving
 - Break of 45 minutes after 6 hours of working

(driving, loading, unloading)

–Daily rest of 11 hours after

*9 hours of driving or

*9.6 hours of working or

*13 hours of wall-clock time after the end of the last daily rest

- Objective function: Minimize sum of fixed and variable costs

Stage 2 problem:

Vehicle routing problem with time windows and multiple depots (VRPTWMD):

- Customers with given time windows and service times, no capacity requirements
- Vehicles stationed at different depots, no restrictions regarding ability to visit customers
- Each vehicle performs at most one route starting and ending at the same specified depot
- Driver rules:
 - Time between two consecutive daily rests less than or equal to 13 hours
 - Route duration less than or equal to the

- maximum time between two weekly rests (144 hours)
- Overall driving time less than or equal to 45 hours
- Overall working time less than or equal to 48 hours
- Objective function: Minimize sum of fixed and variable costs for drivers and shuttles

Note that the maximum route duration in both problems is the same. This is because we assume a planning horizon of one working week as this is common in practice. Theoretically, the planning horizon for the stage 1 problem, where the truck routes are computed, could be chosen arbitrarily, whereas in the stage 2 problem, where the driver routes are determined, the time between two weekly rests is postulated by the law. Note further that it is necessary to consider a period of 13 hours between two daily rests in both problems to ensure that feasible driver routes can be computed in stage 2. This is because in stage 1, the 13-hour rule is only maintained on each truck route segment between two relay stations, whereas in stage 2, the 13-hour rule must also be ensured if a driver changes trucks. Otherwise, it might happen that a driver is assigned two truck route segments of less than 13 hours' duration each without scheduled daily rest. If the two segments have a cumulative duration of more than 13 hours, then not considering the 13-hour rule in stage 2 would lead to an illegal driver route.

In both stages, we apply an implementation of a large neighborhood search heuristic, see [Shaw \(1997\)](#), [Ropke and Pisinger \(2006\)](#), [Pisinger and Ropke \(2007\)](#). Essentially, in such methods, elements of a solution are alternately removed (destruction step) and reinserted (reconstruction step) in order to improve a given solution. In the context of PDPs/VRPs, given a complete route plan, a subset of requests/customers is removed from their respective routes in the destruction step and reinserted into the resulting partial routes in the reconstruction step. The basic steps of LNS in general and our implementation in particular are the following:

Large Neighborhood Search

Construct an initial feasible solution x and save x as the current best solution x^* , i.e., set $x^* := x$

Repeat

Select a destruction and a repair operator

Create a neighboring solution x' from x using the procedures corresponding to the selected destruction and repair operators

Improve the new solution x' by local search

If solution x' can be accepted, i.e., if an acceptance criterion is fulfilled, set $x := x'$

If x is better (has a lower objective function value) than x^* , set $x^* := x$

Until a termination criterion is reached

Return x^*

For the construction of an initial feasible solution as well as for reconstruction purposes, we use a parallel best insertion procedure: In each iteration, for each remaining unplanned request, the best (cheapest) possible insertion positions for the pickup and the delivery location into each existing route are determined. The request whose best possible insertion causes the smallest increase in the costs of the current route plan is then included. A new vehicle is used only if an insertion into existing routes is impossible. Since our SVCRSP considers a heterogeneous fleet, whenever a new route is to be created, all unused vehicles must be checked for suitability. After the recreate step, we use a swap operator, that is, the exchange of two requests between two routes, as a local search improvement step.

We use the following three destruction procedures:

- A request is selected at random and the route that contains this request is destroyed, that is, all requests that are on the same route as the specified request are also selected. This is repeated until a specified percentage of requests has been determined.
- A route ρ_1 is selected at random. This route and the one closest to it, say, route ρ_2 , are destroyed. (A route ρ_2 is closest to ρ_1 if the overall pairwise distance of all locations visited in ρ_1 to the locations visited in ρ_2 is minimal over all routes.) This is repeated until a specified percentage of requests is determined.
- A subset S of requests is selected at random. (The requests need not necessarily be on the same route.) Then, further requests whose pickup or delivery locations are closest to any pickup or delivery location of the requests in S are selected until a specified percentage of requests has been determined.

The selected requests are then removed from their routes, and vehicles serving now empty routes are considered unused. In each iteration, one of the three destruction operators is chosen randomly.

The algorithm accepts every new solution that exceeds the old objective function value, $c(x)$, by less than a given factor $\delta \geq 1$; that is, a new solution x' is accepted if $c(x') < \delta c(x)$. The algorithm terminates after a specified number of iterations (cf. Section 7). Our implementation is described in more detail in Sigl (2009). The design of the code is generic and can easily be adapted, or, rather, instantiated, to solve PDPTWRSs as well as VRPTWMDs.

In our algorithm, there are data structures that store and update the resources of each individual driver throughout the planning horizon to correctly account for each individual driver's driving and working history from the beginning of the planning horizon. Similarly, there are data structures for each individual vehicle, storing all relevant information (costs, capacities, home depot, itinerary, schedule).

6.1 Stage 1: Determination of truck routes

The determination of truck routes in the course of the PDPTWRS in the first stage is based on an appropriate network representation. In addition, the consideration of driver rules and relay stations requires modeling efforts.

6.1.1 Construction of the network

We consider a network $D_1 = (V_1, A_1)$, where V_1 is the set of vertices and A_1 is the set of arcs. The vertex set $V_1 = S \cup E \cup P \cup D \cup R$ consists of vertices for start and end depots in subsets S and E respectively, vertices for pickup and delivery in subsets P and D , and vertices for relay stations in subset R . ($A \cup B$ denotes the union of the disjoint sets A and B .) This means that if a physical depot location may be used as a relay station, this location is represented by three different vertices in the network (one in S , one in E , and one in R). The start of (un)loading at any vertex $i \in V$ must be within a prescribed time window $[a_i, b_i]$. Depot and relay station vertices have time windows starting Monday, 0:00 hours, and ending Saturday, 24:00 hours. For each request, a pickup and a delivery vertex with associated time windows and (un)loading times are given. The (un)loading must

begin, but need not be finished, within the time window. The arc set A_1 is composed as follows:

- (i) From each start depot vertex i , there is one arc to the corresponding end depot vertex (for unused trucks), to each relay station vertex $j \in R$ (since it may be sensible or necessary to visit a relay station before picking up the first request), and to each pickup vertex.
- (ii) There is an arc between each pair of relay station vertices, from each relay station vertex $i \in R$ to every pickup and every delivery vertex, and to each end depot vertex $j \in E$.
- (iii) Finally, there is an arc between each pair of request vertices (except for an arc from the delivery vertex of a request to the corresponding pickup vertex), an arc from each request vertex to each relay station vertex, and there is an arc from each delivery vertex to each end depot vertex.

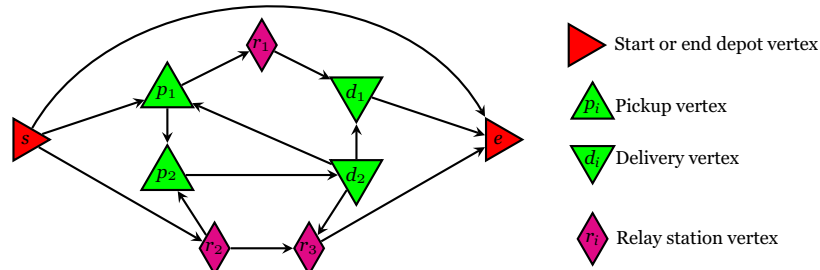
Each arc (i, j) has an associated distance $d_{ij} \geq 0$ and travel time $t_{ij} \geq 0$. These weights correspond to the real-world distance and travel time by truck between the respective locations and are assumed to be equal for each truck type. The travel time indicates the pure driving time, disregarding any breaks or rests. Of course, only arcs (i, j) with $a_i + t_{ij} \leq b_j$ are introduced.

An example network is depicted in Figure 2. The network represents an instance with one depot location, three relay station locations, and two pickup-and-delivery requests. Consequently, the network contains one start depot vertex, one end depot vertex, three relay station vertices, and two pickup and two delivery vertices. To keep the figure clear and concise, there is only one arc for each arc type described in items (i)-(iii). For example, the absence of an arc from vertex p_1 to vertex d_1 does not mean that there is no such arc in the network.

6.1.2 Consideration of driver rules in stage 1

In the PDPTWRS, that is, in the first stage of our solution approach, routes are determined for the trucks. The routes start at the depot vertices and visit pickup and delivery vertices of compatible requests, maintaining the capacity constraints of the trucks, the time windows of the requests, the time period until the next visit at a relay station, and the following driver rules resources: time since the end of the last daily rest, daily and interval driving time, daily and interval working time. A visit at a relay station resets all these driver rules

Figure 2: Example network for stage 1



resources; the other resources are considered in the second stage of our solution procedure.

An important feature of the approach is that the trucks need not be empty when visiting a relay station. Usually, at relay stations, a driver change is performed. For that reason, a truck may leave a relay station after a short constant period of time after having reached the station, a so-called *relay time*. In the computational experiments, this time was set to 15 minutes.

Within our insertion heuristics, insertions are performed with the objective of minimizing costs. For a route, that is, for a given sequence of vertices, determined according to this objective, a schedule considering the above-mentioned driver rules is computed that has a minimum overall route duration under the condition to arrive at the end depot vertex as early as possible. This is sensible when one and the same driver drives the entire route of a truck. For each route, a complete scheduled sequence of activities (drive, wait, (un)load, break, rest) is determined, indicating the earliest sensible beginning of each activity. This means that a concrete departure time from the start vertex of each segment is known.

6.1.3 Procedure for considering relay stations

The procedure for considering relay stations is a straightforward extension of the existing procedure for the PDPTW. In each iteration, after routes have been modified by adding, shifting, or removing requests in the (re)construction or destruction step, a concrete schedule is computed for all modified routes, taking into account driver rules. These routes are then checked for whether they violate the maximum time period between two visits at a relay station (but are otherwise feasible with

respect to time windows, driver rules, and truck capacity).

For a route/vertex sequence $(s, i_1, i_2, \dots, i_m, i_{m+1}, \dots, i_n, e)$, this check works as follows: The route is traversed backward from the end depot vertex $e \in E$. If, according to the route's schedule, the time period between the arrival at e and the departure from the last relay station visited (or the start depot, if no relay station is visited at all) is greater than allowed, a relay station is inserted. Assume, for example, that, in the above route, $i_m \in S \cup P \cup D \cup R$ is the first vertex encountered such that there is no relay station between i_m and e and that the time between the departure from i_m and the arrival at e exceeds the allowed time. Then a relay station vertex $r \in R$ is inserted between i_m and e . (The important question of *which* vertex $r \in R$ to insert is discussed below.) An attempt is made to insert r directly after i_m . If this is not possible (due to time window constraints), insertion is attempted directly after i_{m+1} etc. The resulting route must be feasible with respect to time windows, driver rules *and* time period between the inserted relay station vertex r and the end depot vertex e . To verify this, it is necessary to compute a new schedule for the complete route. If the insertion was successful, the procedure is repeated, now moving backward from the last inserted relay station vertex r toward the start depot vertex s . These steps are iterated until a feasible route and a corresponding schedule are determined or the route is discarded for infeasibility.

If a request is removed from a route in the destruction step of the large neighborhood search, all relay stations visited *after* the pickup vertex of the request are removed, too. This is motivated by the fact that it may be useful to reduce the number

of relay stations, but to keep some relay stations in a partial route that was shown to be feasible. In the computational experiments, also attempts were made to remove all relay stations from a route if at least one request was removed from this route. This increased the running times, but did not lead to better results.

As mentioned, an important issue is the selection of the relay station vertex to be inserted between two vertices. Here, we opted for the station causing the smallest detour. Seen globally, though, this might be suboptimal, since there might be no driver for changing at the relay station selected in this way. There might be more drivers available for continuing the truck route at other, more distant, relay stations. Choosing such a station might help to avoid shuttle transports. Hence, storing the number of drivers available at different relay stations in different time intervals might be sensible. However, such a more sophisticated strategy is highly unlikely to be beneficial: Visiting a more distant relay station to find a suitable truck/driver for continuing the route will obviously increase the truck distance and the driving time compared to visiting the closest station. This means that shuttle distance and travel time are replaced by truck distance and travel time, but truck kilometers are more expensive than shuttle kilometers, and truck hours count as driving time for the driver, whereas shuttle time does not.

6.2 Stage 2: Determination of driver routes

In the second stage, a driver must be found for each truck route segment determined in stage 1. Recall that the truck routes consist of one or more route segments starting and ending at a depot or a relay station. Each segment must by assumption be driven by one and the same driver with one and the same truck. To obtain feasible routes for drivers, the following points must be ensured (since they have not already been ensured in stage 1):

- (a) The time between two consecutive daily rests on a driver route must be less than or equal to 13 hours.
- (b) The overall duration of a driver route must be less than or equal to 144 hours.
- (c) The overall driving (working) time on a driver route must be less than or equal to 45 hours (48 hours).

To achieve this, an appropriate network representation is required.

6.2.1 Construction of the network

The network $D_2 = (V_2, A_2)$ for the VRPTWMD is set up as follows: V_2 consists of one start and one end depot vertex for each home depot of a driver, and of one customer vertex for each segment of a truck route computed in stage 1. For a stage 2 customer vertex i , let $s_i \in S \cup R$ and $e_i \in R \cup E$ denote the start and end vertices of its corresponding stage 1 segment respectively. That is, the stage 1 segment corresponding to stage 2 vertex i starts at location l_{s_i} and ends at location l_{e_i} . All such l_{s_i} and l_{e_i} are either depots or relay stations. Again, the depot vertices have time windows starting Monday, 0:00 hours, and ending Saturday, 24:00 hours. The time window $[a_i, b_i]$ of each customer vertex i is equal to the start time window of the corresponding segment. The determination of these time windows is explained in detail below.

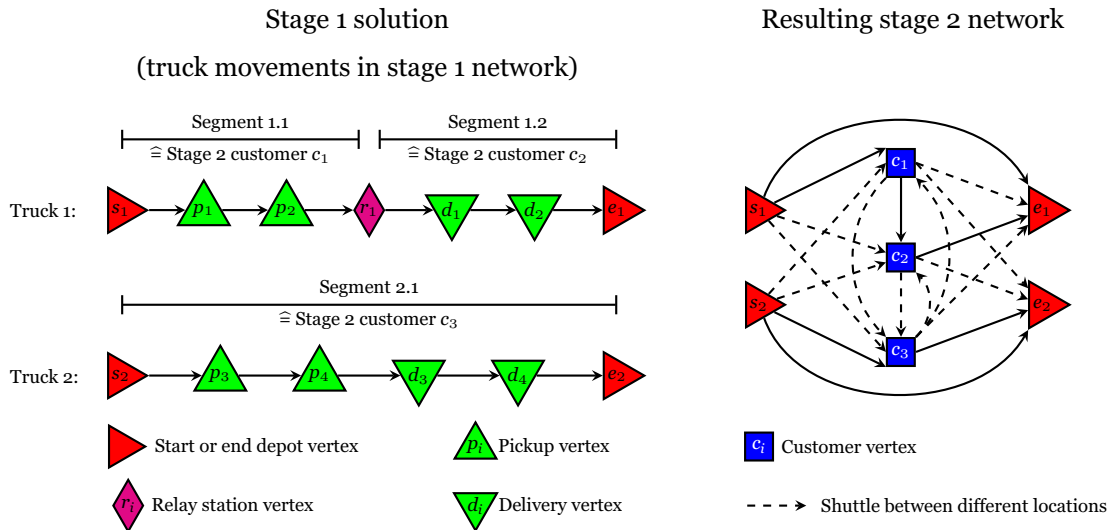
If no shuttle transports are allowed, the arc set A_2 is comprised of the following arcs (where l_i denotes the physical location corresponding to vertex $i \in V_2$):

- (i) From each start depot vertex i , there is one arc to the corresponding end depot vertex (for unused drivers) and to each customer vertex j with $l_i = l_{s_j}$.
- (ii) There is an arc between each pair (i, j) of customer vertices with $l_{e_i} = l_{s_j}$.
- (iii) Finally, there is an arc from each customer vertex i to each end depot vertex j with $l_{e_i} = l_j$.

If shuttle transports are allowed, the $l_i = l_{s_j}$, $l_{e_i} = l_{s_j}$, and $l_{e_i} = l_j$ restrictions are abandoned. That is, if time windows allow, there are arcs from each start depot vertex to each customer vertex, between each pair of customer vertices, and from each customer vertex to each end depot vertex. Figure 3 depicts an example of how a network of a stage 2 instance is constructed from a stage 1 solution.

On the left-hand side of Figure 3, a solution to stage 1 is described that contains two truck routes. The first route starts and ends at depot 1 and consists of two segments. The second route starts and ends at depot 2 and involves one segment. The resulting stage 2 network is depicted on the right-hand side. It is assumed that drivers are available at the same two depots where the trucks are stationed. Therefore, the stage 2 network possesses two start and two end depot vertices. Moreover,

Figure 3: Example network for stage 2



in accordance with the routes from stage 1, the network comprises three customer vertices, one for each route segment. The solid arcs represent possible movements of drivers between vertices corresponding to the same physical location, that is, where no driver shuttle transports are necessary. For example, there is a solid arc from start depot vertex s_1 to customer vertex c_1 , since c_1 corresponds to stage 1 route segment 1.1, which starts at vertex s_1 . The dashed arcs represent driver movements between different physical locations, that is, movements requiring a shuttle transport. The depicted stage 2 network is complete in the sense that all possible arcs resulting from the stage 1 solution are shown. For example, there is no arc from vertex c_2 to c_1 , because using such an arc would imply that segment 1.2, which corresponds to c_2 , is performed before segment 1.1, which corresponds to c_1 . This is impossible, since the requests delivered on segment 1.2 are picked up on segment 1.1. Moreover, depending on the stage 2 time windows (see below), also some of the depicted arcs may be impossible to use.

Each arc $(i,j) \in A_2$ has an associated distance $d_{ij}^{shuttle} \geq 0$ and travel time $t_{ij}^{shuttle} \geq 0$. In the stage 2 network, these weights correspond to the real-world distance and travel time *by shuttle van* between the respective locations. The weights $d_{ij}^{shuttle}$ are needed to determine the shuttle costs, which are assumed to be distance-dependent. For

arcs (i,j) linking vertices corresponding to the same physical location, $d_{ij}^{shuttle} := t_{ij}^{shuttle} := 0$. This means that, if no shuttle transports are allowed, both $d_{ij}^{shuttle}$ and $t_{ij}^{shuttle}$ are zero for all arcs, since in this case, there are only arcs linking vertices corresponding to the same physical location. The travel time $t_{ij}^{shuttle}$ indicates the pure driving time, disregarding any breaks or rests, because driver rules are not relevant for drivers of small shuttle vans. However, working time rules are relevant. To completely account for these, it would be necessary to introduce a third algorithmic stage and compute explicit routes and schedules for shuttle vans and their drivers. This is beyond the scope of the present paper and left as a topic for future research, see the Conclusions. Instead, we assume that there is a sufficient number of shuttle vans with rested shuttle van drivers so that (i) a truck driver never has to wait for a shuttle and (ii) it can be assumed that each shuttle trip between two relay stations requires at most one break of 45 minutes for the shuttle driver, and does so only in the rare case that the pure driving time of the shuttle transport exceeds 6 hours. As shuttle vans have no speed limit on German Autobahns, this break time can be compensated for by going faster than the average speed used by a standard GIS system to compute driving times. In order to meet the above-mentioned three requirements (a)-(c) concerning driver rules, three

additional weights, $w_{ij}^d \geq 0$, $w_{ij}^w \geq 0$, and $t_{ij}^{driver} \geq 0$, are associated with each arc $(i, j) \in A_2$. The weights w_{ij}^d and w_{ij}^w respectively measure the driving and working time accrued on the weekly driving and working time account of the driver who uses arc (i, j) . The weight t_{ij}^{driver} measures the wall-clock time that will have elapsed after arrival at vertex i or after the beginning of i 's time window, whichever is greater, until a driver can reach j via the arc (i, j) .

To illustrate this, the following explanations are appropriate: If a driver uses an arc (i, j) emanating from a customer vertex i and leading to a customer or end depot vertex j , this indicates that the driver drives the stage 1 segment corresponding to customer i . In other words, a 'visit' at a stage 2 customer induces a 'service time' at the customer vertex. This service time is equal to the time for executing the corresponding stage 1 segment. As mentioned above, this time is known from the unequivocal duration determined for each stage 1 route and comprises driving, (un)loading, waiting, break, and rest times along the segment. Driving and (un)loading time count as working time. Moreover, a change of vehicle incurs a relay time, which also counts as working time. (Entering the first vehicle at the beginning of a route and leaving the last vehicle at the end of a route also takes time, but this time is not counted as working time.) In addition, a driver is assumed to take a daily rest at a relay station, which increases overall route duration.

When shuttle transports are allowed, it must be considered that the time a truck driver spends being driven in a shuttle van, although not counted as working time, must also not be counted as break or rest time. Therefore, when a driver reaches a relay station after a shuttle ride, the time since the end of his last daily rest will be positive. To ensure that such a driver can still perform the corresponding segment $i \in V_2$, we take into account the time t_i^{before} that elapses before the beginning of the first daily rest, and the time t_i^{after} that elapses after the end of the last daily rest on i . Both values are known from the computations performed in stage 1. If no daily rest is taken on a segment, we assume that a daily rest is necessary directly before and directly after the segment. In order to ensure that two segments/customers i and j requiring a shuttle transport can be performed consecutively

by any driver (at least as far as the 13-hour rule is concerned), we distinguish three cases:

- (i) If $t_i^{after} + t_{ij}^{shuttle} \leq 13$, the shuttle is carried out directly after performing the segment corresponding to i , and a daily rest is taken at l_{s_j} . (In this case, it is assumed, without loss of generality, that the daily rest lasts at least until the beginning of the time window at j .)
- (ii) If $t_i^{after} + t_{ij}^{shuttle} > 13$ and $t_{ij}^{shuttle} + t_i^{before} \leq 13$, a daily rest is taken at l_{e_i} directly after performing the segment corresponding to i ; subsequently, the shuttle is executed and no daily rest is taken at l_{s_j} . (In this case, it is assumed that the daily rest lasts long enough so that there is no waiting time at j .)
- (iii) If $t_i^{after} + t_{ij}^{shuttle} > 13$ and $t_{ij}^{shuttle} + t_i^{before} > 13$, a daily rest is taken directly before and directly after the shuttle transport. (Again, it is assumed that the daily rest at l_{s_j} lasts at least until the beginning of the time window at j .)

Similar considerations can be applied to potential arcs from a depot vertex to a customer vertex.

Now, let t_i^{drive} and t_i^{work} respectively denote the driving time and working time along the segment corresponding to i , and let $t_i^{overall}$ denote the 'service' time for execution of the segment corresponding to i as explained above. The values of t_i^{drive} and t_i^{work} can easily be determined from the scheduled sequence of activities computed in stage 1 for each truck route. Furthermore, let t^{daily_rest} denote the time for the daily rest(s) between two vertices (11 or 22 hours, depending on which of the above cases (i)-(iii) holds), and let t^{relay} denote the time for handing over a truck to another driver or being handed over a truck from another driver. Then, in view of the above elaborations and the described network structure, w_{ij}^d , w_{ij}^w , and t_{ij}^{driver} are set as follows:

- $w_{ij}^d := w_{ij}^w := t_{ij}^{driver} := 0$, if i is a start depot vertex and j is a customer vertex with $l_i = l_{s_j}$.
- $w_{ij}^d := w_{ij}^w := 0$, and $t_{ij}^{driver} := t_{ij}^{shuttle} + t^{daily_rest}$, if i is a start depot vertex and j is a customer vertex with $l_i \neq l_{s_j}$.
- $w_{ij}^d := t_i^{drive}$, $w_{ij}^w := t_i^{work}$, and $t_{ij}^{driver} := t_i^{overall} + t^{daily_rest}$, if i and j are customer vertices and consecutive segments on one stage 1 route. In this case, no handover time is incurred, since the driver stays on the same vehicle.
- $w_{ij}^d := t_i^{drive}$, $w_{ij}^w := t_i^{work} + 2t^{relay}$, and $t_{ij}^{driver} := t_i^{overall} + t^{daily_rest} + 2t^{relay} + t_{ij}^{shuttle}$, if i and j are

customer vertices and *not* consecutive segments on one stage 1 route.

Here, the first relay time is incurred for handing over the truck k to the driver who will perform the next segment on k 's route. The second relay time is incurred for being handed over the truck k from the driver who has performed the preceding segment on k 's route.

- $w_{ij}^d := t_i^{drive}$, $w_{ij}^w := t_i^{work}$, and $t_{ij}^{driver} := t_i^{overall}$, if i is a customer vertex and j is an end depot vertex with $l_{e_i} = l_j$.
- $w_{ij}^d := t_i^{drive}$, $w_{ij}^w := t_i^{work}$, and $t_{ij}^{driver} := t_i^{overall} + t^{daily_rest} + t_{ij}^{shuttle}$, if i is a customer vertex and j is an end depot vertex with $l_{e_i} \neq l_j$.

Note that the end depot vertex corresponds to the home depot location of the driver, not the truck. The truck is left at location l_{e_i} .

Using these values, only arcs (i, j) with $a_i + t_{ij}^{driver} \leq b_j$ are introduced in order to satisfy the time window constraints.

The observance of 13 hours at most between two daily rests is now ensured by the fact that a driver always takes a daily rest between visiting two customers: At the relay station where the segment corresponding to the first customer ends, at the relay station where the segment corresponding to the second customer starts, or at both locations. Using an arc (i, j) increases the route duration by a wall-clock time of t_{ij}^{driver} . Depending on the actual arrival time at j , waiting time may occur. However, as explained above, it is assumed that this waiting time is added to the daily rest(s) taken between i and j .

The observance of the time since the end of the last weekly rest is ensured by the network structure. Each driver performs at most one route, he is assumed rested upon leaving his start depot vertex (that is, all driver rules resources are reset), and the time windows at the start and end depot vertices prohibit routes longer than 144 hours.

The only checks concerning driver rules that have to be performed *within* the algorithm used to solve the stage 2 problem, no matter which VRPTWMD algorithm is actually used, are the checks whether the accumulated weekly driving and working times, that is, the sum of the w_{ij}^d and the sum of the w_{ij}^w of all arcs on a driver's route, are less than or equal to 45 and 48 hours respectively. Such checks are essentially equivalent to checking the vehicle capacity constraint in a standard VRP, because the

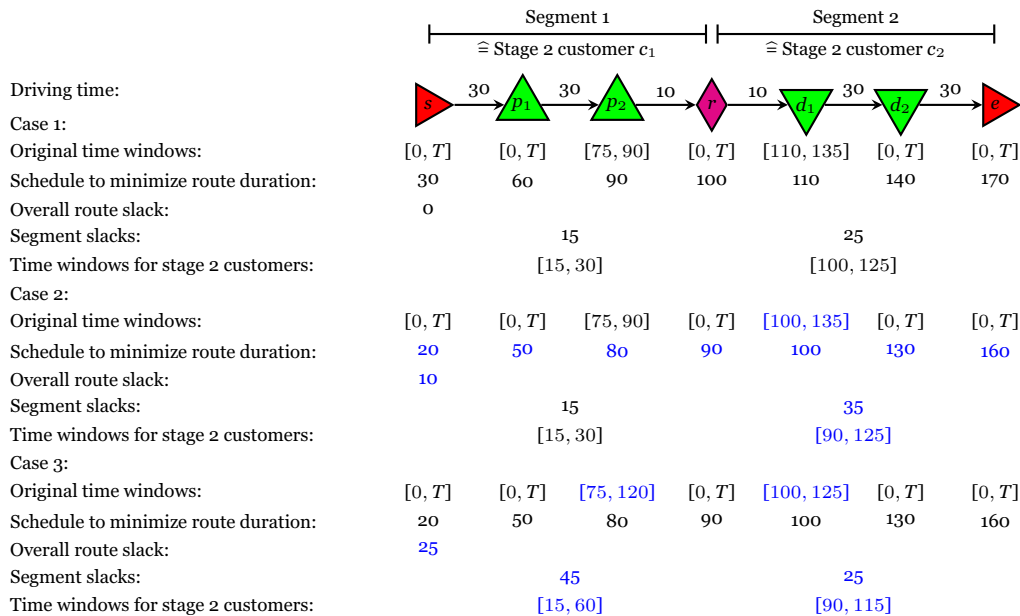
capacity consumption in a standard VRP, though usually associated with a customer vertex, can of course be associated with the arcs emanating from the vertex. For any insertion of a customer into a route, this check can be performed in constant time, independent of the number of customers on the route, cf. [Irnich \(2008\)](#).

6.2.2 Considering time windows and dealing with the interdependence problem

Even after separating the determination of truck and driver routes, the interdependence problem remains. Recall that in the stage 1 routine for determining a schedule considering driver rules, a schedule is computed specifying a concrete departure time from the start vertex of each segment. Now, depending on the request time windows, the driver rules, and the limited time between two visits at a relay station, it may on the one hand be possible to arrive at a relay station vertex sooner than the stage 1 schedule indicates, and it may on the other hand be possible to depart from a relay station vertex later than the stage 1 schedule denotes. In other words, there may be slack within a segment/route. Moreover, the amount of slack may differ between the segments of the same route. As an example, consider Figure 4 and assume, for simplicity, zero (un)loading and relay times and a maximum time between two visits at a relay station of 130 time units (arbitrarily chosen). Driver rules are irrelevant in this example, because the overall route duration is so short that no breaks or rests are necessary or sensible. T denotes the length of the planning horizon.

The route depicted in Figure 4 performs two pickup-and-delivery requests ($p_1 \rightarrow d_1$ and $p_2 \rightarrow d_2$) and consists of two segments. Segment 1 starts at a start depot, picks up the two requests, and ends at a relay station. Segment 2 starts at this relay station, delivers the two requests, and ends at an end depot. With respect to the time windows of the requests, we distinguish three cases. In each case, the line 'Schedule to minimize route duration' indicates the departure time at each vertex so that the overall route duration is minimized under the condition to arrive at the end depot vertex as early as possible. The line 'Overall route slack' specifies by how much the scheduled departure time from the start depot may be shifted without violating any time windows. The line 'Segment slacks' provides this information for each segment. The line

Figure 4: Example for stage 2 time windows



“Time windows for stage 2 customers” indicates the longest possible time windows for the stage 2 customers corresponding to the stage 1 segments, based on the minimum duration schedule and the segment slacks. The durations of the two segments, or, in other words, the service times for the stage 2 customers, are the same in all three cases: They are simply the sum of the segment driving times, that is, 70 time units (since there is no unavoidable waiting time, and since there are no stage 1 (un)loading or relay times).

In case 1, it is easy to see that the time windows for the customers in the stage 2 VRPTWMD induce no interdependencies: If customer c_1 is visited at time 30, that is, as late as possible, then customer c_2 can still be visited at time 100, that is, as early as possible. This is because it is ensured that the stage 1 segment corresponding to c_1 will be finished early enough ($30 + 70 = 100$), so that the truck used to perform both segments will be at the relay station no later than at time 100.

In case 2, if customer c_1 is visited later than at time 20, customer c_2 is affected, and thus the stage 2 routes visiting the customers are interdependent. For example, let c_2 be on route ρ_2 and suppose that, due to subsequent customers, c_2 must be visited no later than at time 99. Further

assume that the route visiting customer c_1 , say, ρ_1 , is changed so that c_1 can be visited no earlier than at time 30. Then, the truck used to perform both segments will not have arrived at the relay station at time 99, and thus ρ_2 becomes infeasible. This, in turn, implies that in stage 2, route changes such as additions, removals, or repositionings of customers may have side effects on other routes, and these effects must be controlled in some way to be able to determine feasible solutions.

In case 3, the situation is still worse. Even if the time windows for the stage 2 customers are not reduced due to insertion into routes, any route visiting c_2 becomes infeasible if c_1 is visited later than at time 45.

To overcome this interdependence problem, there are several possibilities. The simplest option, of course, is to use the given concrete schedule for each route (that is, a time window of length zero for each stage 2 request) and essentially solve a pure vehicle *scheduling* problem. This greatly reduces the optimization potential for stage 2.

The other extreme is to use the slack as described above and, in stage 2, consider the route interdependencies and devise a specific algorithm for solving the VRPTWMD. This preserves the complete optimization potential (as given by the stage 1

solution). It is doubtful, though, whether this second approach is promising for the large instances we try to solve.

We therefore chose a compromise. Using the flexibility given by a stage 1 solution, we set the time windows of the stage 2 requests in such a way that all requests, also those corresponding to segments belonging to the same stage 1 route, are independent of one another. More precisely, we define the slack for each segment as the maximal time by which the departure time from the start vertex of a segment can be deferred without violating any time window within the current segment *and* such that the actual departure of the subsequent segment is not delayed. In this way, no request interdependencies arise in stage 2, but still some flexibility with respect to the time windows of the requests is achieved. For the three cases in Figure 4, the procedure works as follows: In case 1, as stated above, we may set the time windows for the stage 2 requests to [15, 30] and [100, 125] respectively. In case 2, we may determine the time windows to [15, 20] and [90, 125], and, finally, in case 3, to [15, 20] and [90, 115].

6.3 Comments on the solution approach

The consequence of disregarding the optional driver rules is that our algorithm sometimes cannot find a legal schedule for a route, although such a schedule exists. It is guaranteed, though, that our algorithm only returns driver routes for which a legal schedule has been found. If no legal schedule is found for a route (no matter whether this is because there is no legal schedule or because the only legal schedule(s) require(s) exploiting optional rules), the route is discarded.

It is not possible to reverse the two stages and compute driver routes in the first stage and truck routes in the second, unless it is assumed that only empty trucks may reach relay stations. This is because when a driver route is computed in stage 1, it is unknown which locations the driver will have to visit after leaving a relay station, because it is unknown which truck he will use and thus which requests he will have to deliver.

In tramp transportation in practice, the planning situation is often a dynamic one. Not all requests may be known in advance, and it may therefore be necessary to perform rolling horizon planning. In such a situation, only one or a few segments must

be computed in advance for a truck. The procedure in stage 1 remains the same, but the number of requests decreases. Stage 2 becomes the problem of assigning exactly one driver to each segment and at most one segment to each driver. Thus, the overall problem becomes easier.

7 Computational experiments

The present work was motivated by a research project studying the potential of an adoption of the *advanced truckload* business model to Germany and Europe. This business model originated in North America as a reaction to the deregulation of the road transport market and has provided a competitive edge for companies that successfully implemented it, the so-called advanced truckload firms (ATLFs). Details on the ATLF business model can be found in [Walther \(2010\)](#). One pillar of ATLFs is the increased temporal utilization of trucks obtained by allowing drivers to switch vehicles (commonly referred to as *slipseating*). Therefore, to assess the potential of slipseating for long-distance tramp transport in Germany by truck, an extensive real-world data set was provided by our practice partner, a major German freight forwarder. The data set comprises 2,800 pickup-and-delivery requests between 1,975 locations and during a planning horizon of six days from Monday to Saturday, 1,645 trucks and drivers stationed at 43 depots, each of which can also function as relay station, and 157 additional relay station locations. All locations are dispersed over an area of approximately 350,000 km². The fleet consists of two types of trucks differing with respect to capacity and ability to perform certain requests, but with the same time- and distance-dependent costs. Every driver is able to drive any truck, and all drivers receive the same wage. An unlimited number of identical shuttle vans is available at any time at any location. The shuttle vans are assumed to incur distance-dependent costs only. Table 1 specifies some further indicators.

We have no data on driver resources such as driving time during the last week etc. Therefore, we assume that at the beginning of the planning horizon, all drivers have the same average driver rules resources, as specified in Section 6. For our purposes, that is, to identify savings potential by switching drivers compared to a fixed truck-driver assignment, this is a valid approach, because for both

Table 1: Data set indicators

| Indicator | Min. | Avg. | Max. |
|---|-------------|-------------|-------------|
| Length of pickup time window [hh:mm] | 00:00 | 06:39 | 13:20 |
| Length of delivery time window [hh:mm] | 00:00 | 18:00 | 30:25 |
| Time between earliest pickup and latest delivery [hh:mm] | 02:15 | 28:10 | 52:00 |
| Distance between pickup and delivery location [truck-kms] | 1 | 391 | 1,117 |
| Request size [Tons / Loading meters] | 0.1 / 0.4 | 10.8 / 9.4 | 25.0 / 14.6 |
| Vehicle capacity type 1 [Tons / Loading meters] | 22.0 / 15.3 | 22.0 / 15.3 | 22.0 / 15.3 |
| Vehicle capacity type 2 [Tons / Loading meters] | 25.0 / 13.6 | 25.0 / 13.6 | 25.0 / 13.6 |

situations, free as well as fixed truck-driver assignment, the same assumptions regarding driver resources are made.

To test and evaluate our algorithm, we implemented its different modules in Delphi/Object Pascal as an add-on to our LNS code described in Section 6. We used the following settings and parameters: In each destruction step, 5% of requests are removed. Due to the large number of requests, a higher percentage is too time consuming. Initially, only improving solutions are accepted. After 500 iterations without improvement, the δ value for accepting new solutions is set to 1.5 and is then gradually decreased to 1 every 500 iterations without improvement. 12,500 iterations of the large neighborhood search algorithm, that is, 12,500 destruction and 12,500 reconstruction steps, are performed in each stage.

We examined the following 14 scenarios:

- Scenario 1: A fixed truck-driver assignment without the need to visit a relay station. This was used as the baseline scenario, since it reflects current practice.
- Scenarios 2-7: A fixed truck-driver assignment where visiting one of 43, 100, 200 relay stations is required every 37 or 48 hours.
- Scenario 8: A free truck-driver assignment without the need to visit a relay station and with the assumption that a daily rest takes only 15 minutes. This is equivalent to assuming that there are infinitely many drivers available at any point in space and time without incurring any shuttle costs. This scenario yields an upper bound on the possible savings.
- Scenarios 9-14: A free truck-driver assignment where visiting one of 43, 100, 200 relay stations is required every 37 or 48 hours, and where the possibility of driver shuttle transports exists.

The values of 37 and 48 hours for the inter-visit

time (the time between two consecutive visits at a relay station) were chosen for the following reasons. The 37-hour period is motivated by the existing legislation: As mentioned, 13 hours after the end of a daily rest, the next daily rest (of 11 hours) must start. Thus, by ensuring that a truck visits a relay station after 37 hours, it is guaranteed that a driver spends at least every other daily rest at a relay station. A period of 48 hours allows at least two visits per week at a relay station.

The objective was to minimize the sum of overall operating costs, which are comprised of four components:

- Fixed costs for
 - each truck that leaves its home depot (150.06 EUR) and
 - each truck driver who leaves his home depot (175.40 EUR).

According to current practice in long-distance road transport in Germany, these costs are charged in daily rates for each started 24-hour period. For example, if a truck route has a duration of 25 hours, counted from the point in time when the truck leaves the start depot until it returns to the end depot, two daily rates are charged for this truck.

- Distance-dependent costs for
 - each truck-km (0.54 EUR) and
 - each shuttle-km (0.12 EUR).

The first scenario was computed with a basic variant of the stage 1 procedure, where no relay stations are considered. This variant was also used for scenario 8, where, in addition, the daily rest duration was set to 15 minutes. Scenarios 2-7 were tackled using only the stage 1 part of our algorithm; scenarios 9-14 were solved using both described stages. Tables 2 and 3 show the computational results. In the tables, the lines ‘No. days’, ‘No. truck days’, and ‘No. driver days’ indicate the number of daily rates

Table 2: Computational results for fixed truck-driver assignment

| Scenario | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|--|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| No. relay stations | – | 43 | 43 | 100 | 100 | 200 | 200 |
| Inter-visit time | – | 37 | 48 | 37 | 48 | 37 | 48 |
| Absolute values: | | | | | | | |
| No. routes | 729 | 767 | 746 | 768 | 752 | 756 | 758 |
| No. days | 3,419 | 3,413 | 3,424 | 3,456 | 3,429 | 3,410 | 3,457 |
| Kilometers | 1,697,000 | 1,786,000 | 1,752,000 | 1,778,000 | 1,754,000 | 1,752,000 | 1,755,000 |
| Costs | 2,029,000 | 2,075,000 | 2,060,000 | 2,085,000 | 2,063,000 | 2,056,000 | 2,073,000 |
| Relative changes to baseline scenario 1 in percent: | | | | | | | |
| No. routes | 0.00 | 5.21 | 2.33 | 5.35 | 3.16 | 3.70 | 3.98 |
| No. days | 0.00 | –0.17 | 0.15 | 1.08 | 0.30 | –0.28 | 1.10 |
| Kilometers | 0.00 | 5.21 | 3.19 | 4.77 | 3.33 | 3.20 | 3.39 |
| Costs | 0.00 | 2.27 | 1.53 | 2.76 | 1.68 | 1.33 | 2.17 |

charged for trucks and/or drivers respectively.

First and foremost, the computational experiments demonstrated that our algorithm is capable of solving large real-world instances and is able to achieve consistent relevant results. Moreover, the following observations can be made in Tables 2 and 3:

- For fixed truck-driver assignment, the number of routes and the overall distance traveled increase considerably when relay stations have to be visited, compared to the baseline scenario where this is not required. The number of truck days, in contrast, changes only slightly. This implies that the average route duration decreases.
- For free truck-driver assignment, the number of truck routes and truck days decreases significantly in each scenario $s = 8, \dots, 14$, both compared to the baseline scenario 1 as well as to the corresponding scenario with fixed truck-driver assignment (scenario $s - 7$). This implies an increase in the average truck route duration, that is, a better temporal utilization of the trucks, which is the result desired. The overall distance traveled also increases compared to the baseline scenario, but changes only slightly compared to the corresponding scenarios with fixed truck-driver assignment. On the other hand, however, for scenarios 9-14, the numbers of driver routes and driver days rise sharply.
- For free truck-driver assignment, increasing the time between two visits at a relay station from 37 to 48 hours leads to a clear reduction of all indi-

cators, truck and driver routes and days as well as truck and shuttle kilometers, independent of the number of relay stations used.

- As previously mentioned, the stage 1 algorithm always inserts the relay station with the smallest detour between vertices of a truck route. For free truck-driver assignment, it might be promising to use a more sophisticated approach for choosing a relay station, where the chances are higher to find a rested driver who may readily be available to take over the truck. The usefulness of such an approach, however, should have become visible comparing the scenarios with 200 and 43 relay stations, but the results showed no such effects.
- The impressive savings obtained with scenario 8, albeit a theoretical value, demonstrate that the rest requirements for drivers are the bottleneck for an efficient use of trucks: The distance savings compared to the baseline scenario 1 are moderate, but if no daily rests are necessary, the number of trucks and the number of truck days can be reduced dramatically.

In addition, the computational experiments showed the following results:

- An increased number of relay stations showed no effects. The potential benefit of reduced detours to reach one of a larger number of relay stations is compensated by the reduced number of routes visiting the same relay station. Put differently, a network of 43 relay stations may already be

Table 3: Computational results for free truck-driver assignment

| Scenario | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|--|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| No. relay stations | ∞ | 43 | 43 | 100 | 100 | 200 | 200 |
| Inter-visit time | 13 | 37 | 48 | 37 | 48 | 37 | 48 |
| Absolute values: | | | | | | | |
| No. truck routes | 486 | 705 | 669 | 712 | 678 | 694 | 679 |
| No. driver routes | 486 | 968 | 962 | 990 | 970 | 977 | 961 |
| No. truck days | 2,323 | 3,229 | 3,178 | 3,240 | 3,187 | 3,192 | 3,205 |
| No. driver days | 2,323 | 4,068 | 3,988 | 4,156 | 4,000 | 4,120 | 4,006 |
| Truck kilometers | 1,652,000 | 1,793,000 | 1,756,000 | 1,777,000 | 1,761,000 | 1,771,000 | 1,770,000 |
| Shuttle kilometers | 0 | 291,000 | 238,000 | 351,000 | 271,000 | 355,000 | 274,000 |
| Costs | 1,648,000 | 2,200,000 | 2,152,000 | 2,215,000 | 2,162,000 | 2,199,000 | 2,171,000 |
| Relative changes to baseline scenario 1 in percent: | | | | | | | |
| No. truck routes | -33.33 | -3.29 | -8.23 | -2.33 | -7.00 | -4.80 | -6.86 |
| No. driver routes | -33.33 | 32.78 | 31.96 | 35.80 | 33.06 | 34.02 | 31.82 |
| No. truck days | -32.05 | -5.56 | -7.06 | -5.25 | -6.80 | -6.63 | -6.26 |
| No. driver days | -32.05 | 18.99 | 16.65 | 21.56 | 16.99 | 20.50 | 17.17 |
| Truck kilometers | -2.69 | 5.63 | 3.47 | 4.69 | 3.73 | 4.32 | 4.27 |
| Costs | -18.78 | 8.43 | 6.06 | 9.17 | 6.55 | 8.38 | 7.00 |
| Relative changes to corresponding scenario with fixed truck-driver assignment in percent: | | | | | | | |
| No. truck routes | -33.33 | -8.08 | -10.32 | -7.29 | -9.84 | -8.20 | -10.42 |
| No. driver routes | -33.33 | 26.21 | 28.95 | 28.91 | 28.99 | 29.23 | 26.78 |
| No. truck days | -32.05 | -5.40 | -7.20 | -6.26 | -7.07 | -6.37 | -7.28 |
| No. driver days | -32.05 | 19.19 | 16.47 | 20.25 | 16.65 | 20.84 | 15.90 |
| Truck kilometers | -2.69 | 0.40 | 0.27 | -0.08 | 0.39 | 1.09 | 0.85 |
| Costs | -18.78 | 6.02 | 4.47 | 6.24 | 4.80 | 6.96 | 4.73 |

considered sufficiently dense for the considered area.

- Whereas, as stated above, removing the need for daily rests leads to an enormous increase in productivity, increasing the temporal capacity and availability of drivers by extending their weekly driving and working time only marginally improved the results for the scenarios with free truck-driver assignment.
- With free truck-driver assignment, no feasible solutions could be computed when no shuttle transports are allowed: In stage 2, only very few routes were found that ensured the return of drivers and trucks to their home depots at the end of the planning horizon.
- We also tried an inter-visit time of 13 hours,

meaning that every daily rest must be taken at a relay station. This did not work well: A large part of the requests could not be fulfilled. For these requests, the time between earliest pickup and latest delivery is not sufficient to cover the distance between pickup and delivery location plus the detour to and from a relay station.

Whether or not monetary savings are possible depends on the relative values of the fixed costs for trucks and drivers and the distance-dependent costs for trucks and shuttle vans. For the observed sharp increase in the number of necessary drivers, it is clear that no monetary savings are to be expected for firms operating in the notoriously high-wage country Germany, where the daily truck rate is less than the daily driver rate. For the given data

set, the truck fixed costs would have to rise by an order of magnitude to yield overall cost savings. Since the number of available drivers is more than twice the number of truck routes, this sharp increase in the number of necessary drivers came as a surprise. After analyzing the data, we identified two reasons:

- (i) On the one hand, for the given requests, the drivers are based at the wrong depots, so that an allocation of free drivers to truck route segments requires a lot of shuttle transports. (It is interesting to note that, given the truck routes from stage 1, it is possible in scenarios 9-14 to assign the 1,645 drivers to the 43, 100 or 200 relay stations so that shuttle transports are necessary only at the beginning and at the end of each driver's working week. Then, monetary savings are obtained also for our project partner's actual cost structure, which may be considered representative for long-distance transport in Central and Northern Europe. Unfortunately, of course, this is only of theoretical relevance, because the drivers live where they live and cannot be resettled.)
- (ii) On the other hand, given the depot and relay station locations, the requests do not possess a suitable structure with respect to their position in space and time. It is too seldom that a truck is available at a relay station for a driver directly after the end of his daily rest, or, respectively, that a rested driver is available to take over a truck directly after its arrival at a relay station. The main cause for this mismatch is that the requests in our testbed were not acquired in a systematic manner by a central department, as it is done in ATLF firms. The results obtained by [Taylor, Meinert, Killian, and Whicker \(1999\)](#) support this conclusion: For North American long-distance road transport, the authors point out the importance of "regularly scheduled delivery capacity in the form of delivery lanes, hubs and zones which regularize driver tours while providing performance benefits for the carrier."

To relativize the above results, the following remarks must be made. The above results are no proof that no savings are possible through slipseating, and it must be pointed out that a third possible reason for not obtaining monetary savings through slipseating compared to a fixed truck-driver assignment could be a low solution quality due to

insufficient performance of the heuristic. It is conceivable that the observed sharp increase in the number of necessary drivers is partly caused by the solution procedure and is not entirely unavoidable for the data used. Moreover, a more powerful heuristic might also be able to further reduce the number of truck routes in stage 1.

8 Summary and research outlook

This paper has studied a simultaneous vehicle and crew (truck and driver) routing and scheduling problem arising in long-distance road transport. A central aspect of the problem is that the usual assumption in the vehicle routing literature of a fixed assignment of a driver to a truck is abandoned. Instead, truck/driver changes are allowed at geographically dispersed relay stations, and driver shuttle transports between these relay stations are considered. This leads to interdependencies between trucks and drivers and requires the synchronization of their routes.

A heuristic solution algorithm based on large neighborhood search has been described. The algorithm proceeds in two stages, computing routes for trucks in stage 1 and routes for drivers in stage 2, where the truck routes from stage 1 constitute the customers/requests to perform in the stage 2 problem. The interdependencies between the truck and the driver routes are mitigated by specifying limited time windows for the truck routes.

Extensive computational experiments have been performed with real-world data provided by a major freight forwarder. The results show the validity of our algorithm. Surprisingly, for typical request structures in non-timetabled long-distance transport in Central Europe, truck/driver changes seem to offer no savings potential. Under the given circumstances, a fixed truck-driver assignment seems to be the right set-up. The lack of savings potential through slipseating is at least partially caused by the lack of appropriate follow-up requests that are adequately situated in space and time, although it cannot be excluded that an insufficient performance of the heuristic algorithm with respect to solution quality is one reason for these results. In any case, it must be noted that the results are only valid for the considered business field. The developed algorithm for simultaneous vehicle and crew routing and scheduling may well lead to very different results with other data or in other application

areas, so that its further study is justified.

A possible improvement of the algorithm is the addition of a third stage, in which routes for shuttle vans are computed. Planning these transports can be done independently of stages 1 and 2, since the stage 2 solution can be used as an input for such a stage 3 procedure in the same manner as stage 1 yields the input for stage 2 in the presented algorithm.

Moreover, as discussed above, it would be highly interesting, though highly difficult, to fathom the potential offered by considering the complete temporal flexibility of the routes computed in stage 1 of the algorithm and develop a procedure capable of dealing with interdependent requests in stage 2. An even more involved extension of our SVCRSP is to allow that a change of driver/truck at a relay station is performed *without* the driver taking a daily rest before switching to another truck. In this case, the interdependence of drivers and trucks becomes still closer, and the described decomposition approach no longer works, which leads to a yet more complicated situation. [Bürckert, Fischer, and Vierke \(2000\)](#), [Drexel \(2007\)](#), and [Cheung, Shi, Powell, and Simão \(2008\)](#), in the context of long-distance road transport, short-distance collection traffic, and seaport container drayage respectively, consider such problems: Composite objects consisting of two or more types of elementary autonomous and/or non-autonomous objects are required to fulfill tasks, and the elementary objects may join and separate on the fly at many different locations. In addition, whereas in our SVCRSP as well as in the references discussed in Section 3, it was always clear and fixed that two types of objects must visit a location simultaneously, the latter three papers offer different options as to which combinations of object type are used to fulfill a task.

Finally, it is sometimes allowed in practice to transship partial and complete loads from one vehicle to another at specified transfer locations, cf. [Bock \(2010\)](#). Studying a problem where drivers are allowed to switch vehicles *and* where requests may be transshipped between vehicles constitutes a challenging area for future research.

Acknowledgement. Michael Drexel was funded by the Deutsche Forschungsgemeinschaft (DFG) under grant no. IR 122/5-1. The authors are grateful to the department editor and two anonymous

referees for their detailed and helpful comments and suggestions.

Appendix: Details on driver rules

Regulation (EC) No. 561/2006 of the European Parliament and of the Council of 15 March 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 specifies the following rules as quoted from [Drexel and Prescott-Gagnon \(2010\)](#):

- Interval driving time
 A driver must not drive for more than 4 hours and 30 minutes consecutively.
- Daily driving time
 The driving time between two daily rest periods (see below) must not exceed 10 hours. At most twice in a calendar week, the driving time between two daily rest periods may exceed 9 hours.
- Weekly driving time
 The driving time within a calendar week must not exceed 56 hours.
- Fortnightly driving time
 The driving time in two consecutive calendar weeks must not exceed 90 hours.
- Breaks
 After 4 hours and 30 minutes of driving, the driver must take a break of at least 45 minutes. The break may be split into two interruptions of at least 15 and at least 30 minutes in this order.
Note The above rule does *not* mean that there can be at most 4 hours and 30 minutes of driving in each interval of 5 hours and 15 minutes. Consider the following example. It is legal for a driver to start his working day by driving for 1 hour and 30 minutes, then take a break of 15 minutes, drive for an additional 3 hours, take a break of 30 minutes, and then drive for another 4 hours and 30 minutes. At the end of these 4 hours and 30 minutes of driving, the driver has been driving for 7 hours and 30 minutes in the last 8 hours.
- Daily rest periods
 There are two different rules determining when a daily rest period is necessary:
 - Within 24 hours after the end of the last daily or weekly rest period, there must be a daily rest period of at least 9 hours.
 - Additionally, when the daily driving time is

exhausted, a daily rest period must be taken before the driver may continue driving. Between two weekly rest periods, there may be at most three daily rest periods of less than 11 hours. A daily rest period may be split into two periods of at least 3 hours and at least 9 hours in this order.

- Weekly rest periods
 There are also two different rules determining when a weekly rest period is necessary:
 - After at most 144 hours after the end of a weekly rest period, there must be an uninterrupted weekly rest period of at least 24 hours. If the duration of the weekly rest period is less than 45 hours, the difference between 45 hours and the actual duration of the weekly rest period must be added as an additional uninterrupted rest period to an uninterrupted rest period of at least 9 hours within three calendar weeks after the calendar week where the thus shortened weekly rest period ends.
 - If the weekly or fortnightly driving time is exhausted, a rest must be taken until the end of the current calendar week, even if this is longer than 45 hours.

In addition, there are rules for vehicles manned by two drivers.

Directive 2002/15/EC of the European Parliament and of the Council of 11 March 2002 on the organisation of the working time of persons performing mobile road transport activities specifies the following rules:

- Interval working time
 Any mobile worker or self-employed driver performing mobile road transport activities must not work for more than 6 hours consecutively.
- Daily working time
 If, within a 24-hour period, work lasting at least 4 hours is to be performed between 00:00 hours and 07:00 hours, the working time within this 24-hour period must not exceed 10 hours.
- Weekly working time
 The working time within a calendar week must not exceed 60 hours. This is allowed only if, over four months, the working time per calendar week does not exceed 48 hours on average.
- Breaks
 After 6 hours of working, the worker must take a break of at least 30 minutes. If the daily working time exceeds 9 hours, the break must be at least 45 minutes. The break may be split into periods

of at least 15 minutes each.

These lists may be incomplete. Other national legislation in Germany or legislation in other countries need not correspond completely to these rules and might be stricter. In addition, other international rules and regulations may apply.

In their paper on driver rules, [Drexl and Prescottt-Gagnon \(2010\)](#) make the following statement: “This is an OR paper, not a juristic text. It is explicitly stated that none of the algorithms presented in this paper is guaranteed to determine a ‘legal schedule’ for a vehicle route, because there is no precise mathematical definition of the term ‘legal schedule’. The determination of a legal schedule for a vehicle route is not a question of mathematics or computer science, but solely a juristic one. The pertinent regulations give abundant room for interpretation, so that any dispute concerning the legality of a route will eventually have to be settled in court.” We agree.

References

- Bock, Stefan (2010): Real-Time Control of Freight Forwarder Transportation Networks by Integrating Multimodal Transport Chains, *European Journal of Operational Research*, 200 (3): 733-746.
- Bürckert, Hans-Jürgen, Klaus Fischer, and Gero Vierke (2000): Holonic Transport Scheduling with Teletruck, *Applied Artificial Intelligence*, 14 (7): 697-725.
- Caprara, Alberto, Leo Kroon, Michele Monaci, Marc Peeters, and Paolo Toth (2007): Passenger Railway Optimization, in: Cynthia Barnhart and Gilbert Laporte (eds.): *Transportation, Vol. 14 of Handbooks in Operations Research and Management Science*, Elsevier: Amsterdam, 129-187.
- Cheung, Raymond, Ning Shi, Warren Powell, and Hugo Simão (2008): An Attribute-Decision Model for Cross-Border Drayage Problem, *Transportation Research Part E*, 44 (2): 217-234.
- Desaulniers, Guy and Mark Hickman (2007): Public Transit, in: Cynthia Barnhart and Gilbert Laporte (eds.): *Transportation, Vol. 14 of Handbooks in Operations Research and Management Science*, Elsevier: Amsterdam, 69-127.
- Drexl, Michael (2007): On Some Generalized Routing Problems, PhD Thesis, Faculty of Business and Economics, RWTH Aachen University.
- Drexl, Michael (2012): Synchronization in Vehicle Routing: A Survey of VRPs with Multiple Synchronization Constraints, *Transportation Science*, 46 (3): 297-316.
- Drexl, Michael and Eric Prescottt-Gagnon (2010): Labelling Algorithms for the Elementary Shortest Path Problem with Resource Constraints Considering EU Drivers’ Rules, *Logistics Research*, 2 (2): 79-96.
- Fügenschuh, Armin (2006): The Vehicle Routing Problem with Coupled Time Windows, *Central European Journal of Operations Research*, 14 (2): 157-176.
- Goel, Asvin (2009): Vehicle Scheduling and Routing with



- Drivers' Working Hours, *Transportation Science*, 43 (1): 17-26.
- Goel, Asvin (2010): Truck Driver Scheduling in the European Union, *Transportation Science*, 44 (4): 429-441.
- Golden, Bruce, S. Raghavan, and Edward Wasil (eds.) (2008): *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer: New York et al., NY.
- Hollis, B., M. Forbes, and B. Douglas (2006): Vehicle Routing and Crew Scheduling for Metropolitan Mail Distribution at Australia Post, *European Journal of Operational Research*, 173 (1): 133-150.
- Irnich, Stefan (2008): Resource Extension Functions: Properties, Inversion, and Generalization to Segments, *OR Spectrum*, 30 (1): 113-148.
- Kim, Byung-In, Jeongin Koo, and Junhyuk Park (2010): The Combined Manpower-Vehicle Routing Problem for Multi-Staged Services, *Expert Systems with Applications*, 37 (12): 8424-8431.
- Klabjan, Diego (2005): Large-Scale Models in the Airline Industry, in: Guy Desaulniers, Jacques Desrosiers, and Marius Solomon (eds.) *Column Generation*, Springer: New York et al., NY, 163-195.
- Kok, A., C. Meyer, H. Kopfer, and J. Schutten (2010): A Dynamic Programming Heuristic for the Vehicle Routing Problem with Time Windows and European Community Social Legislation, *Transportation Science*, 44 (4): 442-454.
- Laurent, Benoît and Jin-Kao Hao (2007): Simultaneous Vehicle and Driver Scheduling: A Case Study in a Limousine Rental Company, *Computers & Industrial Engineering*, 53 (3): 542-558.
- Lenstra, Jan-Karel and Alexander Rinnooy Kan (1981): Complexity of Vehicle Routing and Scheduling Problems, *Networks*, 11 (2): 221-227.
- Pisinger, David and Stefan Ropke (2007): A General Heuristic for Vehicle Routing Problems, *Computers & Operations Research*, 34 (8): 2403-2435.
- Prescott-Gagnon, Eric, Guy Desaulniers, Michael Drexler, and Louis-Martin Rousseau (2010): European Driver Rules in Vehicle Routing with Time Windows, *Transportation Science*, 44 (4): 455-473.
- Prescott-Gagnon, Eric, Guy Desaulniers, and Louis-Martin Rousseau (2010): Heuristics for an Oil Delivery Vehicle Routing Problem, Working Paper, Les Cahiers du GERAD G-2010-63.
- Ropke, Stefan and David Pisinger (2006): An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows, *Transportation Science*, 40 (4): 455-472.
- Shaw, Paul (1997): A New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems, Working Paper, University of Strathclyde.
- Sigl, Thomas (2009): Modellbasierte Entscheidungsunterstützung am Beispiel des Verkehrsplanungsframeworks *CG_{frame}*, in: Stefanie Müller, Angela Roth, and Norbert Schmidt (eds.): *Märkte, Anwendungsfelder und Technologien in der Logistik*, Gabler: Wiesbaden, 361-378.
- Solomon, Marius (1987): Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints, *Operations Research*, 35 (2): 254-265.
- Taylor, G., T. Meinert, R. Killian, and G. Whicker (1999): Development and Analysis of Alternative Dispatching Methods in Truckload Trucking, *Transportation Research Part E*, 35 (3): 191-205.
- Toth, Paolo and Daniele Vigo (eds.) (2002): *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics: Philadelphia, PA.
- Walther, Stefan (2010): *Industrializing Transportation Networks*, Shaker: Aachen.
- Xiang, Zhihai, Chengbin Chu, and Haoxun Chen (2006): A Fast Heuristic for Solving a Large-Scale Static Dial-A-Ride Problem under Complex Constraints, *European Journal of Operational Research*, 174 (2): 1117-1139.
- Zäpfel, Günther and Michael Bögl (2008): Multi-Period Vehicle Routing and Crew Scheduling with Outsourcing Options, *International Journal of Production Economics*, 113 (2): 980-996.

Biographies

Michael Drexler works in the field of applied operational research in logistics and transport and is currently engaged in a project on synchronization in vehicle routing at the Chair of Logistics Management at Johannes Gutenberg University in Mainz.

Julia Rieck studied Applied Mathematics at the University of Göttingen and the University of Hamburg. She obtained her PhD in 2008 and now holds the position of a researcher and teaching assistant at Clausthal University of Technology. Her research interests include vehicle routing and location problems, distribution network design, and project scheduling problems.

Thomas Sigl studied Computer Sciences at the Friedrich-Alexander-University Erlangen-Nürnberg. He worked for 10 years at the Fraunhofer Centre for Applied Research on Supply Chain Services SCS, Department Networks. His research interests include vehicle routing problems and logistics network design.

Bettina Press studied Mathematics at the University of Erlangen-Nürnberg. Since 2010 she works at the Fraunhofer Centre for Applied Research on Supply Chain Services SCS, and since 2011 she is head of the Optimization group at SCS. Her research interests include network design and vehicle routing.