Bidirectional labeling for solving vehicle routing and truck driver scheduling problems *

Christian Tilk
Johannes Gutenberg University Mainz, Mainz, Germany
tilk@uni-mainz.de

Asvin Goel
Kühne Logistics University, Hamburg, Germany
asvin.goel@the-klu.org

This paper studies the vehicle routing and truck driver scheduling problem where routes and schedules must comply with hours of service regulations for truck drivers. It presents a backward labeling method for generating feasible schedules and shows how the labels generated with the backward method can be combined with labels generated by a forward labeling method. The bidirectional labeling is embedded into a branch-and-price-and-cut approach and evaluated for hours of service regulations in the United States and the European Union. Computational experiments show that the resulting bidirectional branch-and-price-and-cut approach is significantly faster than unidirectional counterparts and previous approaches.

Keywords: Routing, Hours of service regulations, truck driver scheduling, bidirectional labeling, branch-and-price-and-cut

---

1. Introduction

In long-distance haulage, truck drivers must comply with hours of service regulations mandating minimal requirements concerning breaks and rest periods. This paper studies the *vehicle routing and truck driver scheduling problem (VRTDSP)* which is a variant of the well-known vehicle routing problem with time windows in which hours of service regulations must be complied with.

Approaches for solving vehicle routing and truck driver scheduling problems have to ensure that all truck drivers take regular breaks and rest periods as mandated by respective hours of service regulations. This is usually achieved by evaluating routes using forward labeling methods, where labels represent possible states of a truck driver after conducting some sequence of activities. These activities include breaks, rest periods, driving periods, and other periods in which the driver is working. For each activity conducted by the truck driver, the label is updated using a so-called *resource extension function (REF)*. Determining a feasible truck driver schedule for a given route

* This paper extends and replaces a previous unpublished working paper which focused on hours of service regulations in the United States.

is a computationally expensive task, because for most hours of service regulations studied in the literature, no polynomial complexity bound is known and the number of route evaluations is usually huge.

In classical vehicle routing the evaluation of routes requires little computational effort. Forward labeling methods, which extend a label from one customer to another with a simple REF, can often be easily turned into backward labeling methods by simply reversing the orientation of the arcs on which the REF is applied (see Figure 1). For cumulative constraints, e.g., capacity constraints, the forward REF f_{nm} and the backward REF g_{nm} can extend the respective label attributes in an identical way. For other constraints, e.g., time windows, label attributes can often be extended by f_{nm} and g_{nm} in a very similar way, i.e, the backward REF is a simple inversion of the forward REF (see Irnich 2008). This ease of reversing the direction allows the use of bidirectional labeling approaches (Righini and Salani 2006). The benefit of bidirectional approaches is that forward and backward labels do not need to be propagated for the entire route. Instead, forward and backward labels can be propagated only up to a so-called half-way point, thus limiting the overall number of labels created and reducing the respective computational burden.

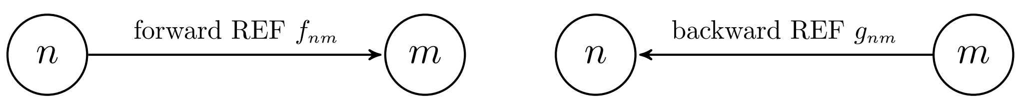

Figure 1 **Forward and backward labeling in classical vehicle routing**

If hours of service regulations must be considered, a single REF between a pair of customers n and m does not suffice because the different driver activities, such as driving periods, breaks, and rests must be explicitly modeled and require dedicated REFs. Furthermore, due to the asymmetry of hours of service regulations, forward and backward labeling methods cannot use the same REFs for forward and backward label propagation. Up to now, it was impossible to leverage the potential of bidirectional approaches for the VRTDSP because it was unclear how to design a backward labeling method for route evaluations subject to hours of service regulations.

This paper presents backward labeling methods for the *US truck driver scheduling problem (US-TDSP)* and the *EU truck driver scheduling problem (EU-TDSP)*, which are the problems of determining a sequence of driver activities allowing a truck driver to visit all customer locations in a route within given time windows and without violating hours of service regulations in the United States (US) and the European Union (EU). We show how backward labels can be combined with forward labels and present a bidirectional labeling method for the US- and EU-TDSP.

The bidirectional labeling method can be used within heuristic and exact approaches for solving the VRTDSP. In heuristic approaches, routes are usually modified using neighborhood operators which make minor changes to one or several routes. Here, solution approaches solely based on forward labeling methods do have a significant disadvantage, because whenever a route is changed, feasibility of the new route can only be validated after calculating new labels for all customer locations in the route from the first change until the end of the route. The bidirectional approach presented in this paper allows to re-evaluate a modified route by only updating labels locally, i.e., from the first change until the last change in the route and by merging forward and backward labels accordingly. This can speed up the evaluation of modified routes significantly.

Exact approaches for vehicle routing problems are often based on *column generation* (CG, see Desaulniers et al. 2005) where new routes are generated solving a *shortest path problem with resource constraints* (SPPRC, see Irnich and Desaulniers 2005). The bidirectional approach presented in this paper can be used to speed up the solution process for the SPPRC. We present such an approach, more precisely, we present a *branch-and-price-and-cut (BPC)* algorithm for the solution of the VRTDSP. Computational experiments demonstrate that bidirectional labeling significantly speeds up the solution process.

Another noteworthy contribution of this paper is that our bidirectional labeling approach is very effective when minimizing cost functions containing a duration-related component. As Regulation (EC) No 561/2006 demands that transport companies do not give drivers any payment related to distances traveled, realistic cost models for European transport operators must consider labour costs that are not based on distances. A realistic model of costs in the EU considers both distance-related costs, such as fuel costs, as well as duration-related costs, such as labor costs. In order to calculate the realistic costs of a route, a truck driver schedule with minimal duration must be found. Finding such a truck driver schedule can be very time-consuming in the presence of hours of service regulations and previous unidirectional approaches (Goel 2018) have performed very poorly in this regard. The BPC algorithm with bidirectional labeling presented in this paper is particularly well suited for such problems and can be used to minimize the weighted sum of distance- and duration-related costs.

The remainder of this paper is as follows: Section 2 briefly reviews the related literature. In Section 3, hours of service regulations in the US and EU are summarized. Section 4 presents the bidirectional labeling approach for the US- and EU-TDSP. We propose new backward labeling methods and show how forward and backward labels can be combined to find feasible truck driver schedules. In Section 5, we present our BPC algorithm for the exact solution of the VRTDSP and show how the bidirectional labeling approach can be used to solve the arising subproblem which is an SPPRC considering hours of service regulations. Section 6 presents computational results for the BPC algorithm and concluding remarks are given in Section 7.

2. Related work

The performance of exact and heuristic algorithms for solving vehicle routing problems strongly depends on the effort required to evaluate the feasibility of routes or partial routes and their contribution to the objective function. For many constraints typically found in vehicle routing problems, route evaluations can be done very efficiently (Vidal et al. 2014, Campbell and Savelsbergh 2004). However, in the presence of hours of service regulations, determining whether a truck driver schedule complying with the regulations exists, is a non-trivial and time-consuming task. Archetti and Savelsbergh (2009) were the first to present a polynomial time approach for checking compliance of a route with US hours of service regulations at that time. Goel and Kok (2012) show that the problem can be solved in $O(k^2)$ time, where k is the number of locations visited by the route. However, with the change in regulations in 2013, these approaches have become obsolete. So far no polynomial complexity bound is known for route evaluation subject to hours of service regulations in the United States and the European Union (Goel 2014, 2010). Furthermore, when searching for feasible truck driver schedules with minimal duration, the computational effort is significantly higher (Goel 2012).

Early approaches for solving vehicle routing problems in the presence of hours of service regulations tried to reduce the computational effort by using simple heuristics for route evaluations (Xu et al. 2003, Zäpfel and Bögl 2008). For example, Xu et al. (2003) evaluate routes by iterating over possible starting times at the first location and determining a unique schedule for each starting time by following the constraints imposed by U.S. hours of service regulations. Among the feasible schedules for the different starting times, the schedule with the smallest costs is selected. Goel (2009) proposes a forward labeling method for EU hours of service regulations considering alternative break and rest schedules and show that significantly better solutions can be found compared to using simple heuristics. Similarly, Kok et al. (2010) present a forward labeling method for EU hours of service regulations considering additional provisions of the regulations. As forward labeling algorithms suffer from the effect that progressively more labels are created, extended, and needed to be stored when the length of the routes increases, Kok et al. (2010) propose to restrict the number of alternative labels to be considered by constant values. Prescott-Gagnon et al. (2010) propose to reduce the computational effort of route evaluations by determining lower and upper bounds on label attributes using forward and backward approaches. A heuristic labeling algorithm is then used to determine a feasible schedule within the tightened bounds. Besides using heuristic labeling methods and constraints on the number of alternative labels, Goel and Vidal (2014) use lower bounds on the duration to travel a given distance in order to avoid the extension of labels to customers that cannot be reached within their time windows. The approaches presented by Goel (2009), Kok et al. (2010), and Prescott-Gagnon et al. (2010) focus on minimizing the total distance whereas Goel and Vidal (2014) minimize the duration of each route only in a post-processing step. Rancourt et al. (2013) propose a forward labeling approach where schedule durations are considered and heuristic dominance rules are used to speed up the solution process.

For a rich vehicle routing problem with simplified break and rest requirements, Ceselli et al. (2009) present a bidirectional dynamic programming approach capable of finding optimal solutions for small scale instances. The first exact approach for hours of service regulations in the United States and the European Union is presented by Goel and Irnich (2017). Goel (2018) extends the unidirectional approach of Goel and Irnich (2017) to accommodate for additional national rules within the European Union and duration-related costs. Although the algorithm does not manage to find optimal solutions for many of the 25 customer instances, the best solutions found within the runtime limit of one hour demonstrated that cost savings can be significant when using realistic cost functions based on distance and duration.

3. Hours of service regulations

This section describes the most important rules of hours of service regulations in the United States and the European Union for a planning horizon of one week.

3.1. United States

Hours of service regulations United States are imposed by the Federal Motor Carrier Safety Administration (2011). According to these regulations, a driver must not drive for more than 11 hours without taking a rest period of at least 10 consecutive hours. The regulation prohibits a driver from driving after 14 hours have elapsed since the end of the last rest period. Furthermore, no driving is allowed if 8 hours have elapsed since the end of the last rest or break period of at least 30 minutes. Lastly, drivers may not be on duty for more than 60 hours in 7 days or 70 hours in 8 days.

Hours of service regulations in the United States do not constrain how drivers are financially compensated and most truck drivers are paid by how many miles they have driven (Bureau of Labor Statistics, U.S. Department of Labor 2019).

3.2. European Union

In the European Union, working hours of truck drivers are governed by Regulation (EC) No 561/2006 and the national implementations of Directive 2002/15/EC (see Goel 2018). Both rule sets impose minimum requirements concerning break periods which must be taken for recuperation. According to Regulation (EC) No 561/2006, a break is an uninterrupted period of at least 45 minutes during which the driver may not carry out any work. Alternatively, a break can be taken in two parts, the first of which must be a period of at least 15 minutes and the second part must be a period of at least 30 minutes. Regulation (EC) No 561/2006 demands that a driver does not drive for more than $4\frac{1}{2}$ hours without taking a break. Directive 2002/15/EC furthermore demands that a driver does not work for more than six hours without a break of at least 30 minutes, and not for more than nine hours without a break of at least 45 minutes. Any break required by Regulation (EC) No 561/2006 also fulfills the break requirements of Directive 2002/15/EC.

Regulation (EC) No 561/2006 requires that a driver takes a rest period of at least 11 hours duration, after an accumulated driving time of at most nine hours. Similar to breaks, a rest period may be replaced by two periods of which the first must take at least three hours and the second at least nine hours. The required rest must be fully taken within 24 hours after the end of the previous rest period. Furthermore, Directive 2002/15/EC limits the amount of daily work if night work is performed. The precise definition of night time and the respective daily working time limit is provided in national law implementing Directive 2002/15/EC (see Goudswaard et al. 2006). Most member states of the EU have night time definitions of four or seven hours duration starting between 20.00h and midnight and ending between 4.00h and 7.00h. A night time definition from 20.00h to 7.00h covers all of the different night time definitions in the EU. In all member states of the EU, the daily working time limit that applies for drivers performing night work is significantly smaller than the amount of work that can legally be conducted during a day. Therefore, we assume in the remainder that drivers take a rest in every night and do not perform night work.

The amount of driving and the amount of working within a week is restricted to at most 56 and 60 hours, respectively.

Regulation (EC) No 561/2006 prohibits transport companies do give drivers any payment related to distances traveled. Consequently, labor costs can only be related to time, for example, by assuming a fix salary per working day (Goel 2018).

3.3. Overview of parameters

The main parameters of hours of service regulations in the United States and the European Union are summarized in Table 1.

Notation	US	EU	Description
t^{rest}	10h	11h	The minimum duration of a rest period
$t^{\mathrm{rest}\vert 1\mathrm{st}}$	-	3h	The minimum duration of the first part of a rest period taken in two parts
$t^{\mathrm{rest}\vert 2\mathrm{nd}}$	-	9h	The minimum duration of the second part of a rest period taken in two parts
t^{break}	$\frac{1}{2}$h	$\frac{3}{4}$h	The minimum duration of a break
$t^{\mathrm{break}\vert 1\mathrm{st}}$	-	$\frac{1}{4}$h	The minimum duration of the first part of a break taken in two parts
$t^{\mathrm{break}\vert 2\mathrm{nd}}$	-	$\frac{1}{2}$h	The minimum duration of the second part of a break taken in two parts
$t^{\mathrm{drive}\vert R}$	10h	9h	The daily driving time limit
$t^{\mathrm{drive}\vert B}$	-	$4\frac{1}{2}$h	The maximum driving time without a break
$t^{\mathrm{work}\vert B}$	-	6h	The maximum amount of work time without a break
$t^{\mathrm{drive}\vert W}$	-	56h	The maximum amount of driving time between weekly rest periods
$t^{\mathrm{work}\vert W}$	60h or 70h	60h	The maximum amount of working time between weekly rest periods
$t^{\mathrm{elapsed}\vert B}$	8h	-	The maximum time after the end of the last break or rest period until which a driver may drive
$t^{\mathrm{elapsed}\vert R}$	14h	-	The maximum time after the end of the last rest period until which a driver may drive
t^{day}	-	24h	The duration of a day
t^{night}	-	4h - 11h	The duration of the time considered as night time
t^{dusk}	-	20h - 24h	The time of the day at which night time begins
t^{dawn}	-	4h - 7h	The time of the day at which night time ends

Table 1 Parameters of hours of service regulations in United States and the European Union

4. Bidirectional labeling

Optimizing vehicle routes subject to hours of service regulations requires a methodology to validate compliance of all routes with the regulations. Furthermore, if total costs are related to the schedule duration, for example, if labor costs are related to time, the cost of performing a route can only be determined if all activities conducted by the truck driver and their durations are known.

The problem of validating compliance of a given route with hours of service regulation is a *truck driver scheduling problem (TDSP)* which is the problem of determining a sequence of driver activities allowing a truck driver to visit a given sequence of locations $(n_1, n_2, \ldots, n_k)$ in such a way that the cumulative duration of all driving activities between each pair of locations n_i and n_{i+1} for $1 \leq i < k$ matches the given driving time $d_{n_i,n_{i+1}}$, that at each location n_i for $1 \leq i \leq k$ a stationary activity of a given duration s_{n_i} is conducted and begins within a given time window $[t^{\min}_{n_i}, t^{\max}_{n_i}]$, and that the sequence of all driver activities complies with applicable hours of service regulations.

Forward and backward labeling methods can be used to modify appropriate labels along a trip between customers n and m using the networks and REFs illustrated in Figure 2. In forward labeling, a REF f^{trip}_{nm} is used to initialize a label attribute for the required driving time of the trip, and a REF f^{visit}_{m} is used to schedule the stationary work at customer m. The REFs f^{a}_{Δ} are used to schedule all driver activities along the trip, where a indicates a parameter for the type of the activity, such as driving, other work, breaks and rests, and Δ indicates a parameter indicating the duration of the activity. For details concering the definition of forward labels and REFs, the reader is referred to Goel and Irnich (2017) and Goel (2018). In backward labeling we can similarly define REFs for backward label propagation. Unlike in classical vehicle routing, backward REFs g^a_Δ for vehicle routing and truck driver scheduling may substantially differ from their forward counterparts f^a_Δ.

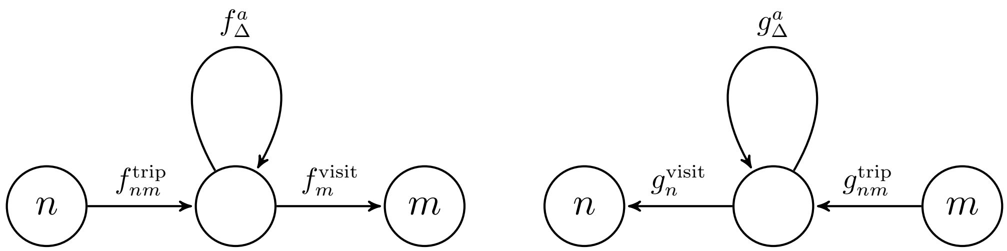

Figure 2 **Forward and backward labeling in vehicle routing and truck driver scheduling**

The difficulty in developing efficient labeling methods for vehicle routing and truck driver scheduling stems from the fact that it is often not possible to decide on the best driver activity a to be conducted next. In the remainder of this section, we present backward labelling methods for hours of service regulations in the United States and the European Union and show how labels generated with forward and backward methods can be combined.

4.1. United States

Before presenting a backward labeling method for hours of service regulations in the United States, let us illustrate the asymmetry of the regulations on a simple example. Figures 3a and 3b show two symmetric schedules that could be obtained by forward and backward approaches.

In the schedule illustrated in Figure 3a, the driver starts fully rested with a work activity for loading the vehicle. After 7 hours of driving, a break is required because a driver is not allowed to drive if a total of 8 hours have elapsed without a break or rest. After the break, the driver continues driving for another 4 hours, after which the destination is reached. Due to strict time requirements on the time of loading and unloading activities in this example, the driver must wait for 3 hours before unloading the vehicle. This schedule complies with hours of service regulations in the United States.

The schedule illustrated in Figure 3b is the symmetric counterpart of the schedule illustrated in Figure 3a. Again, the driver starts fully rested with a work activity for loading the vehicle. After 3 hours of waiting, the driver drives for 4 hours before a break is required. After the break, the driver continues driving for another 7 hours before unloading the vehicle. This schedule, however, violates hours of service regulations in the United States, because the driver is driving after 14 hours have elapsed without a rest.

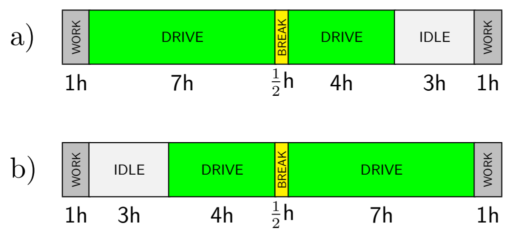

Figure 3 **Symmetric schedules**

Given the asymmetry of the regulations, we cannot simply invert the forward REFs of Goel and Irnich (2017) when developing a backward method. This section describes a backward labeling method for the US-TDSP. For the sake of conciseness, the following presentation focuses on the description of the backward labeling method and details concerning the underlying reasoning are provided in the Appendix. Similar, to the forward labeling method by Goel and Irnich (2017), the backward labeling method represents the state of the driver by a multi-dimensional label $l_\triangleleft$. The index $\triangleleft$ is used to highlight that the label belongs to the backward method. Similarly, we will later use an index $\triangleright$ indicating labels belonging to a forward method. The attributes of a backward label

$$l_\triangleleft = (l_\triangleleft^{\text{time}}, l_\triangleleft^{\text{trip}}, l_\triangleleft^{\text{work|W}}, l_\triangleleft^{\text{drive|R}}, l_\triangleleft^{\text{elapsed|R}}, l_\triangleleft^{\text{elapsed|B}}, l_\triangleleft^{\text{earliest|R}}, l_\triangleleft^{\text{earliest|B}})$$

can be interpreted as follows:

- $l_\triangleleft^{\text{time}}$ represents the start time of the earliest activity,
- $l_\triangleleft^{\text{trip}}$ represents the remaining driving time on the trip to the previous customer,
- $l_\triangleleft^{\text{work|W}}$ represents the accumulated working time,
- $l_\triangleleft^{\text{drive|R}}$ represents the accumulated driving time preceding the next rest,
- $l_\triangleleft^{\text{elapsed|R}}$ represents the time elapsed until the end of the last driving activity preceding the next rest,
- $l_\triangleleft^{\text{elapsed|B}}$ represents the time elapsed until the end of the last driving activity preceding the next break or rest,
- $l_\triangleleft^{\text{earliest|R}}$ represents the earliest possible time at which the last driving activity preceding the next rest must be completed,
- $l_\triangleleft^{\text{earliest|B}}$ represents the earliest possible time at which the last driving activity preceding the next break or rest must be completed.

A label representing the state of a driver who ends service at location n_k is

$$l_\triangleleft = (t^{\max}_{n_k}, 0, s_{n_k}, 0, 0, 0, -\infty, -\infty).$$

This label can be used in a labeling method as an initial label which is changed using the REFs presented below.

Resource extension functions. Given a backward label representing the driver state at location m, the possible driver states at location n can be calculated by finding a path through the backward network shown in Figure 2. The path begins with a link from location m to an intermediate vertex along which a REF g^{trip}_{nm} is used to set the required driving time between n and m. Then, the path can continue along the loops and the REFs g^{drive}_Δ, g^{rest}_Δ, g^{break}_Δ, and g^{idle}_Δ are used to update the driver state depending on the duration Δ of the respective driver activity. Eventually, the path continues from the intermediate vertex to location n and REF g^{visit}_n is used to update the label to consider the stationary work conducted at that location. Table 2 shows how label attributes $l_\triangleleft$ are updated to $\hat{l}_\triangleleft$ by the REFs related to driving and other work. Blank entries indicate that the resource value is kept.

$\hat{l}_\triangleleft$	$g^{\text{trip}}_{nm}(l_\triangleleft)$	$g^{\text{drive}}_\Delta(l_\triangleleft)$	$g^{\text{visit}}_n(l_\triangleleft)$
$\hat{l}^{\text{time}}_\triangleleft$		$l^{\text{time}}_\triangleleft - \Delta$	$\min\{l^{\text{time}}_\triangleleft - s_n, t^{\max}_n\}$
$\hat{l}^{\text{trip}}_\triangleleft$	d_{nm}	$l^{\text{trip}}_\triangleleft - \Delta$	
$\hat{l}^{\text{work\|W}}_\triangleleft$		$l^{\text{work\|W}}_\triangleleft + \Delta$	$l^{\text{work\|W}}_\triangleleft + s_n$
$\hat{l}^{\text{drive\|R}}_\triangleleft$		$l^{\text{drive\|R}}_\triangleleft + \Delta$	
$\hat{l}^{\text{elapsed\|R}}_\triangleleft$		$l^{\text{elapsed\|R}}_\triangleleft + \Delta$	$\max\{l^{\text{elapsed\|R}}_\triangleleft + s_n, l^{\text{earliest\|R}} - t^{\max}_n\}$ (R)
$\hat{l}^{\text{elapsed\|B}}_\triangleleft$		$l^{\text{elapsed\|B}}_\triangleleft + \Delta$	$\max\{l^{\text{elapsed\|B}}_\triangleleft + s_n, l^{\text{earliest\|B}}_\triangleleft - t^{\max}_n\}$ (B)
$\hat{l}^{\text{earliest\|R}}_\triangleleft$			$\max\{l^{\text{earliest\|R}}_\triangleleft, t^{\min}_n + l^{\text{elapsed\|R}}_\triangleleft + s_n\}$ (R)
$\hat{l}^{\text{earliest\|B}}_\triangleleft$		$\max\{l^{\text{earliest\|B}}_\triangleleft, l^{\text{earliest\|R}}_\triangleleft - t^{\text{elapsed\|R}} + l^{\text{elapsed\|B}}_\triangleleft + \Delta\}$	$\max\{l^{\text{earliest\|B}}_\triangleleft, t^{\min}_n + l^{\text{elapsed\|B}}_\triangleleft + s_n\}$ (B)

(R): Label attribute is only updated if $l^{\text{elapsed|R}}_\triangleleft > 0$
(B): Label attribute is only updated if $l^{\text{elapsed|B}}_\triangleleft > 0$

Table 2 Backward REFs related to driving and other work

The REF g^{trip}_{nm} initializes the required driving time for the trip to d_{nm} and leaves all other label attributes unchanged. The REF g^{drive}_Δ reduces the time attribute and the remaining driving time by the duration of the driving activity Δ. Similarly, the cumulative amount of on-duty time, the driving time preceding the next rest, and the elapsed time until the end of the last driving period preceding the next break and rest are increased by Δ. Furthermore, it increases the earliest possible completion time of the last driving period preceding the next break if necessary. The REF g_n^{visit} decreases the time attribute to the latest time at which the stationary work at customer n can be started. Note that service at a location n must start inside but can end outside the time window. If $l_\triangleleft^{\text{time}} > t_n^{\max} + s_n$, then idle time is inserted after the last driving period preceding the next break or rest period so that the time elapsed does not have to be increased or is increased to the smallest possible value. The REF increases the cumulative amount of on-duty time by the duration of the stationary work s_n. If $l_\triangleleft^{\text{elapsed|R}} > 0$, the time elapsed as well as the earliest completion time of the last driving period preceding the next rest are increased if the time window requires this. Similarly, if $l_\triangleleft^{\text{elapsed|B}} > 0$ the respective label attributes are updated.

Table 3 shows how the REFs related to off-duty periods update label attributes. All these REFs

$\hat{l}_\triangleleft$	$g_\Delta^{\text{rest}}(l_\triangleleft)$	$g_\Delta^{\text{break}}(l_\triangleleft)$	$g_\Delta^{\text{idle}}(l_\triangleleft)$
$\hat{l}_\triangleleft^{\text{time}}$	$l_\triangleleft^{\text{time}} - \Delta$	$l_\triangleleft^{\text{time}} - \Delta$	$l_\triangleleft^{\text{time}} - \Delta$
$\hat{l}_\triangleleft^{\text{drive\|R}}$	0		
$\hat{l}_\triangleleft^{\text{elapsed\|R}}$	0	$l_\triangleleft^{\text{elapsed\|R}} + \Delta\ (R)$	$l_\triangleleft^{\text{elapsed\|R}} + \Delta\ (R)$
$\hat{l}_\triangleleft^{\text{elapsed\|B}}$	0	0	$l_\triangleleft^{\text{elapsed\|B}} + \Delta\ (B)$
$\hat{l}_\triangleleft^{\text{earliest\|R}}$	$-\infty$		
$\hat{l}_\triangleleft^{\text{earliest\|B}}$	$-\infty$	$-\infty$	

(R): Label attribute is only updated if $l_\triangleleft^{\text{elapsed|R}} > 0$
(B): Label attribute is only updated if $l_\triangleleft^{\text{elapsed|B}} > 0$

Table 3 Backward REFs related to off-duty periods

reduce the time attribute by the duration of the off-duty period. The REF g_Δ^{rest} sets the cumulative values of driving, as well as the time elapsed until the completion of the last driving periods before the next rest and break to zero. Also the earliest completion times of these driving periods is set to $-\infty$. The REF g_Δ^{break} sets the time elapsed until the completion of the last driving period before the next break to zero and the earliest completion time of this driving period is set to $-\infty$. The time elapsed until the end of the last driving period preceding the next rest is increased if necessary. Similarly, REF g_Δ^{idle} increases the time elapsed until the end of the last driving periods preceding the next rest or break if necessary.

Feasibility conditions. Whether $g^a_\Delta(l_\triangleleft)$ complies with US hours of service regulations can be determined based on the attribute values of $l_\triangleleft$. In order to only generate labels complying with the regulations, the feasibility conditions given in Table 4 must be satisfied when using the corresponding REFs.

REF	Feasibility conditions
$g^{\text{drive}}_\Delta(l_\triangleleft)$	$\Delta \leq \Delta^{\texttt{US}}_{l_\triangleleft}$
$g^{\text{visit}}_n(l_\triangleleft)$	$l^{\text{trip}}_\triangleleft = 0,\ l^{\text{time}}_\triangleleft \geq t^{\min}_n + s_n,\ l^{\text{work}\mid\text{W}}_\triangleleft + s_n \leq t^{\text{work}\mid\text{W}},\ (l^{\text{elapsed}\mid\text{R}}_\triangleleft = 0$ or $\max\{l^{\text{elapsed}\mid\text{R}}_\triangleleft + s_n, l^{\text{earliest}\mid\text{R}}_\triangleleft - t^{\max}_n\} \leq t^{\text{elapsed}\mid\text{R}}),\ (l^{\text{elapsed}\mid\text{B}}_\triangleleft = 0$ or $\max\{l^{\text{elapsed}\mid\text{B}}_\triangleleft + s_n, l^{\text{earliest}\mid\text{B}}_\triangleleft - t^{\max}_n\} \leq t^{\text{elapsed}\mid\text{B}})$
$g^{\text{rest}}_\Delta(l_\triangleleft)$	$\Delta \geq t^{\text{rest}}$
$g^{\text{break}}_\Delta(l_\triangleleft)$	$\Delta \geq t^{\text{break}},\ l^{\text{elapsed}\mid\text{R}}_\triangleleft + \Delta \leq t^{\text{elapsed}\mid\text{R}}$
$g^{\text{idle}}_\Delta(l_\triangleleft)$	$l^{\text{elapsed}\mid\text{R}}_\triangleleft + \Delta \leq t^{\text{elapsed}\mid\text{R}},\ l^{\text{elapsed}\mid\text{B}}_\triangleleft + \Delta \leq t^{\text{elapsed}\mid\text{B}}$

Table 4 Feasibility conditions in backward labeling

REF g^{drive}_Δ is feasible if and only if the duration of the driving period does not exceed the largest possible driving time given by

$$\Delta^{\texttt{US}}_{l_\triangleleft} := \min\{l^{\text{trip}}_\triangleleft, t^{\text{work}\mid\text{W}} - l^{\text{work}\mid\text{W}}_\triangleleft, t^{\text{drive}\mid\text{R}} - l^{\text{drive}\mid\text{R}}_\triangleleft, t^{\text{elapsed}\mid\text{R}} - l^{\text{elapsed}\mid\text{R}}_\triangleleft, t^{\text{elapsed}\mid\text{B}} - l^{\text{elapsed}\mid\text{B}}_\triangleleft\}. \tag{1}$$

REF g^{visit}_n is feasible if and only if location n is reached, the time allows a visit within the time window, the cumulative on-duty time is not exceeded, and no subsequent driving is conducted after 8 or 14 hours after the last break or rest. REF g^{rest}_Δ is feasible if and only if the rest has the required duration. REF g^{break}_Δ is feasible if and only if the break has the required duration and no subsequent driving is conducted after 14 hours have elapsed without a rest. Similarly, REF g^{idle}_Δ is feasible if and only if no subsequent driving is conducted after 14 hours have elapsed without a rest or 8 hours have elapsed without a break.

Dominance. We can use dominance rules to reduce the number of alternative labels to be considered. Given two feasible labels $l_\triangleleft$ and $\bar{l}_\triangleleft$ which both represent a driver state at the beginning of the partial route $(n_i, n_{i+1}, \ldots, n_k)$ with $1 \leq i \leq k$, we write $l_\triangleleft \preceq \bar{l}_\triangleleft$ if $l^j_\triangleleft \leq \bar{l}^j_\triangleleft$ for all $j \in \{\text{trip}, \text{work}\mid\text{W}, \text{drive}\mid\text{R}, \text{elapsed}\mid\text{R}, \text{elapsed}\mid\text{B}, \text{earliest}\mid\text{R}, \text{earliest}\mid\text{B}\}$ and $l^{\text{time}}_\triangleleft \geq \bar{l}^{\text{time}}_\triangleleft$. If $l_\triangleleft \preceq \bar{l}_\triangleleft$, then we also have $g(l_\triangleleft) \preceq g(\bar{l}_\triangleleft)$ for each REF $g \in \{g^{\text{trip}}_{nm}, g^{\text{drive}}_\Delta, g^{\text{rest}}_\Delta, g^{\text{break}}_\Delta, g^{\text{idle}}_\Delta, g^{\text{visit}}_n\}$ because all REFs are non-decreasing in all resources. Hence, $l_\triangleleft$ dominates $\bar{l}_\triangleleft$ and $\bar{l}_\triangleleft$ can be discarded from the set of labels to be updated.

In the Appendix it is shown which sequences of driver activities are dominated by others. Based on these findings we can find conditions telling us when a REF is inferior to another. For this we extend our labels by an additional attribute $l_{\triangleleft}^{\text{last}}$ indicating the last activity scheduled. Table 5 provides an overview of inferiority conditions.

REF	Inferiority conditions
$g_{\Delta}^{\text{idle}}(l_{\triangleleft})$	always
$g_{\Delta}^{\text{drive}}(l_{\triangleleft})$	$\Delta < \Delta_{l_{\triangleleft}}^{\texttt{US}}$
$g_{\Delta}^{\text{rest}}(l_{\triangleleft})$	$\Delta_{l_{\triangleleft}}^{\texttt{US}} > 0$ or $\Delta > t^{\text{rest}}$ or $l_{\triangleleft}^{\text{last}} \in \{g^{\text{rest}}, g^{\text{break}}\}$ or $(l_{\triangleleft}^{\text{trip}} = 0, l_{\triangleleft}^{\text{time}} \leq t_n^{\max} + s_n)$
$g_{\Delta}^{\text{break}}(l_{\triangleleft})$	$\Delta_{l_{\triangleleft}}^{\texttt{US}} > 0$ or $\Delta > t^{\text{break}}$ or $l_{\triangleleft}^{\text{last}} \in \{g^{\text{rest}}, g^{\text{break}}\}$ or $(l_{\triangleleft}^{\text{trip}} = 0, l_{\triangleleft}^{\text{time}} \leq t_n^{\max} + s_n)$

Table 5 Inferiority conditions in backward labeling

If any of these conditions hold, the backward label obtained by applying the REF would be dominated and can thus be omitted. The conditions tell us that it is never beneficial to explicitly schedule idle periods, that it is always better to schedule driving activities as long as possible, and that it is always better to schedule break and rest periods as short as possible. Also a break or rest should not be scheduled if the last activity was a break or rest or if it is possible to schedule the next visit.

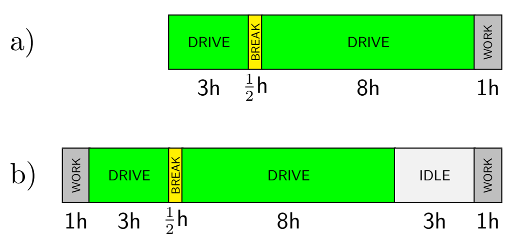

Figure 4 Compliant backward schedules

Figure 4 gives an example of how a schedule for the example in Figure 3 can be found using the backward REFs. The schedule illustrated in Figure 4a gives a partial schedule obtained by $\hat{l}_{\triangleleft} = g_3^{\text{drive}} \circ g_{\frac{1}{2}}^{\text{break}} \circ g_8^{\text{drive}} \circ g_m^{\text{visit}}(l_{\triangleleft})$ where $l_{\triangleleft}$ indicates the initial driver state. Exploiting our inferiority conditions, we only used the maximum possible driving time and minimum allowed break duration. The schedule illustrated in Figure 4b is obtained by $g_n^{\text{visit}}(\hat{l}_{\triangleleft})$. To avoid excessive time elapsed, the REF g_n^{visit} inserts idle time after the last driving period, eventually producing a compliant schedule for our example. Note how the schedule shown in Figure 4b differs from the schedule shown Figure 3a, which would be obtained by a forward method.

Combining forward and backward labels. Following Goel and Irnich (2017), a forward label can be represented by

$$l_\triangleright = (l_\triangleright^{\text{time}}, l_\triangleright^{\text{trip}}, l_\triangleright^{\text{work|W}}, l_\triangleright^{\text{drive|R}}, l_\triangleright^{\text{elapsed|R}}, l_\triangleright^{\text{elapsed|B}}, l_\triangleright^{\text{latest|R}}, l_\triangleright^{\text{latest|B}})$$

where the label attributes can be interpreted as follows:

- $l_\triangleright^{\text{time}}$ represents the completion time of the latest activity,
- $l_\triangleright^{\text{trip}}$ represents the remaining driving time on the trip to the next customer,
- $l_\triangleright^{\text{work|W}}$ represents the accumulated working time,
- $l_\triangleright^{\text{drive|R}}$ represents the accumulated driving time since the last rest,
- $l_\triangleright^{\text{elapsed|R}}$ represents the time elapsed since the end of the previous rest,
- $l_\triangleright^{\text{elapsed|B}}$ represents the time elapsed since the end of the previous break or rest,
- $l_\triangleright^{\text{latest|R}}$ represents the latest possible time at which the previous rest must be completed,
- $l_\triangleright^{\text{latest|B}}$ represents the latest possible time at which the previous break or rest must be completed.

The US-TDSP for a given route $(n_1, n_2, \ldots, n_k)$ can now be solved by determining forward labels for a partial route $(n_1, n_2, \ldots, n_i)$ and backward labels for a partial route $(n_i, n_{i+1}, \ldots, n_k)$ and checking for a feasible combination for each pair of forward and backward labels.

We now show how a forward label $l_\triangleright$ associated to a driver state upon completion of a partial route $(n_1, n_2, \ldots, n_i)$ can be combined with a backward label $l_\triangleleft$ associated to a driver state when beginning a partial route $(n_i, n_{i+1}, \ldots, n_k)$. Note, that both the forward and the backward labeling method add the stationary work at location n_i. Thus, we need to be careful that we do not double count the respective duration s_{n_i} when checking whether the respective schedules can be combined.

A forward label $l_\triangleright$ and a backward label $l_\triangleleft$ at the same location n_i can be combined if

$$l_\triangleright^{\text{time}} - l_\triangleleft^{\text{time}} \leq s_{n_i} \tag{2a}$$

$$l_\triangleright^{\text{work|W}} + l_\triangleleft^{\text{work|W}} - s_{n_i} \leq t^{\text{work|W}} \tag{2b}$$

$$l_\triangleright^{\text{drive|R}} + l_\triangleleft^{\text{drive|R}} \leq t^{\text{drive|R}} \tag{2c}$$

$$l_\triangleright^{\text{elapsed|R}} + l_\triangleleft^{\text{elapsed|R}} \leq t^{\text{elapsed|R}} + s_{n_i} \text{ or } l_\triangleright^{\text{elapsed|R}} \leq s_{n_i} \text{ or } l_\triangleleft^{\text{elapsed|R}} \leq s_{n_i} \tag{2d}$$

$$l_\triangleright^{\text{elapsed|B}} + l_\triangleleft^{\text{elapsed|B}} \leq t^{\text{elapsed|B}} + s_{n_i} \text{ or } l_\triangleright^{\text{elapsed|B}} \leq s_{n_i} \text{ or } l_\triangleleft^{\text{elapsed|B}} \leq s_{n_i} \tag{2e}$$

$$-l_\triangleright^{\text{latest|R}} + l_\triangleleft^{\text{earliest|R}} \leq t^{\text{elapsed|R}} \tag{2f}$$

$$-l_\triangleright^{\text{latest|B}} + l_\triangleleft^{\text{earliest|B}} \leq t^{\text{elapsed|R}} \tag{2g}$$

Analogously, if above conditions hold for $g_{t^{\text{rest}}}^{\text{rest}}(l_\triangleleft)$ or $g_{t^{\text{break}}}^{\text{break}}(l_\triangleleft)$, then it is possible to combine the forward schedule with the backward schedule obtained by adding a rest or break.

4.2. European Union

The asymmetry of EU hours of service regulations mainly results from the possibility of taking breaks and rests in two parts with different durations. A backward label can be represented by

$$l_{\triangleleft} = (l_{\triangleleft}^{\text{time}}, l_{\triangleleft}^{\text{trip}}, l_{\triangleleft}^{\text{work|W}}, l_{\triangleleft}^{\text{drive|W}}, l_{\triangleleft}^{\text{drive|R}}, l_{\triangleleft}^{\text{work|B}}, l_{\triangleleft}^{\text{drive|B}}, l_{\triangleleft}^{\text{elapsed|R}}, l_{\triangleleft}^{\text{earliest|R}}, l_{\triangleleft}^{\text{rest}}, l_{\triangleleft}^{\text{break}}, l_{\triangleleft}^{\text{days}}, l_{\triangleleft}^{\text{dawn}})$$

where $l_{\triangleleft}^{\text{time}}$, $l_{\triangleleft}^{\text{trip}}$, $l_{\triangleleft}^{\text{work|W}}$, and $l_{\triangleleft}^{\text{drive|R}}$ have the same interpretation as for US hours of service regulations. The remaining label attributes can be interpreted as follows:

- $l_{\triangleleft}^{\text{drive|W}}$ represents the accumulated driving time,
- $l_{\triangleleft}^{\text{work|B}}$ represents the accumulated working time preceding the next break or rest,
- $l_{\triangleleft}^{\text{drive|B}}$ represents the accumulated driving time preceding the next break or rest,
- $l_{\triangleleft}^{\text{elapsed|R}}$ represents the time elapsed until the point in time where the next rest is fully taken,
- $l_{\triangleleft}^{\text{earliest|R}}$ represents the earliest possible time at which the next rest must be fully taken,
- $l_{\triangleleft}^{\text{rest}}$ represents the amount of rest required, i.e., zero or $t^{\text{rest|1st}}$, and
- $l_{\triangleleft}^{\text{break}}$ represents the amount of break time required, i.e., zero or $t^{\text{break|1st}}$,
- $l_{\triangleleft}^{\text{days}}$ represents the number of days until the end of the last activity,
- $l_{\triangleleft}^{\text{dawn}}$ represents the time at which the previous night ends.

For a driver who ends service on any given day of the planning horizon, we have to distinguish between the cases that the driver has already taken the first part of a rest period, or not. In the first case, 9 hours of rest are required after the last activity. In the second case, 11 hours of rest are required. In a backward labeling method, these two cases result in different initial labels with different values of $l_{\triangleleft}^{\text{elapsed|R}}$, $l_{\triangleleft}^{\text{earliest|R}}$, and $l_{\triangleleft}^{\text{rest}}$.

For a planning horizon covering several full days and a location n_k with $s_{n_k} = 0$ and a time window spanning the full planning horizon, a backward label representing the state of a driver who ends service on the jth day with an 11 hour rest is

$$l_{\triangleleft} = \big((j-1)\cdot t^{\text{day}} + t^{\text{dusk}}, 0, 0, 0, 0, 0, 0, t^{\text{rest}}, (j-1)\cdot t^{\text{day}} + t^{\text{dawn}} + t^{\text{rest}}, 0, 0, 1, (j-1)\cdot t^{\text{day}} + t^{\text{dawn}}\big).$$

A backward label representing the state of a driver who ends service on the jth day with a 9 hour rest can be obtained by changing above label by setting $l_{\triangleleft}^{\text{elapsed|R}} = t^{\text{rest|2nd}}$, $l_{\triangleleft}^{\text{earliest|R}} = (j-1)\cdot t^{\text{day}} + t^{\text{dawn}} + t^{\text{rest|1st}} + t^{\text{rest|2nd}}$, and $l_{\triangleleft}^{\text{rest}} = t^{\text{rest|1st}}$. If $s_{n_k} > 0$ or the time window of location n_k is narrower, the initial labels can be adjusted accordingly.

For each day in the planning horizon and the two cases, the respective labels can be used as initial labels of a backward labeling method using the REFs presented below.

Resource extension functions. Analogously to the case of US hours of service regulations, we can propagate backward labels along the arcs of the network shown in Figure 2. Given the differences in the regulations we need dedicated REFs g_Δ^{drive}, $g_\Delta^{\text{dayrest}}$, $g_\Delta^{\text{dayrest|2nd}}$, $g_\Delta^{\text{nightrest}}$, $g_\Delta^{\text{nightrest|2nd}}$, $g_\Delta^{\text{rest|1st}}$, $g_\Delta^{\text{break|2nd}}$, $g_\Delta^{\text{break|1st}}$, and g_Δ^{idle} to update the driver state depending on the duration Δ of the respective driver activity. Note, that we explicitly distinguish between a full rest and the second part of a rest taken in two parts and rest periods taken during day time and those covering a night. Furthermore, we only consider breaks taken in two parts because the full break requirement can also be fulfilled by taking both parts immediately after another.

Table 6 shows how label attributes are updated by the REFs related to driving and other work. These REFs are very similar to those for US hours of service regulations with differences in determining $l_\triangleleft^{\text{elapsed|R}}$ and $l_\triangleleft^{\text{earliest|R}}$.

$\hat{l}_\triangleleft$	$g_{nm}^{\text{trip}}(l_\triangleleft)$	$g_\Delta^{\text{drive}}(l_\triangleleft)$	$g_n^{\text{visit}}(l_\triangleleft)$
$\hat{l}_\triangleleft^{\text{time}}$		$l_\triangleleft^{\text{time}} - \Delta$	$\min\{l_\triangleleft^{\text{time}} - s_n, t_n^{\max}\}$
$\hat{l}_\triangleleft^{\text{trip}}$	d_{nm}	$l_\triangleleft^{\text{trip}} - \Delta$	
$\hat{l}_\triangleleft^{\text{drive\|W}}$		$l_\triangleleft^{\text{drive\|W}} + \Delta$	
$\hat{l}_\triangleleft^{\text{drive\|R}}$		$l_\triangleleft^{\text{drive\|R}} + \Delta$	
$\hat{l}_\triangleleft^{\text{drive\|B}}$		$l_\triangleleft^{\text{drive\|B}} + \Delta$	
$\hat{l}_\triangleleft^{\text{work\|W}}$		$l_\triangleleft^{\text{work\|W}} + \Delta$	$l_\triangleleft^{\text{work\|W}} + s_n$
$\hat{l}_\triangleleft^{\text{work\|B}}$		$l_\triangleleft^{\text{work\|B}} + \Delta$	$l_\triangleleft^{\text{work\|B}} + s_n$
$\hat{l}_\triangleleft^{\text{elapsed\|R}}$		$l_\triangleleft^{\text{elapsed\|R}} + \Delta$	$\max\{l_\triangleleft^{\text{elapsed\|R}} + s_n, l_\triangleleft^{\text{earliest\|R}} - t_n^{\max}\}$
$\hat{l}_\triangleleft^{\text{earliest\|R}}$		$\max\{l_\triangleleft^{\text{earliest\|R}}, l_\triangleleft^{\text{dawn}} + l_\triangleleft^{\text{elapsed\|R}} + l_\triangleleft^{\text{rest}} + l_\triangleleft^{\text{break}} + \Delta\}$	$\max\{l_\triangleleft^{\text{earliest\|R}}, t_n^{\min} + l_\triangleleft^{\text{elapsed\|R}} + s_n\}$

Table 6 Backward REFs related to driving and other work

Table 7 shows how the REFs related to rest periods taken during day time update label attributes. These REFs reduce the current time and reset the cumulative values concerning driving and working. All of the REFs reduce the time attribute by Δ, set the cumulative values of driving and working since the last break or rest and the time elapsed since the last rest to zero. The REFs $g_\Delta^{\text{dayrest}}$ and $g_\Delta^{\text{dayrest|2nd}}$ update the time elapsed until the point in time where the rest is fully taken and $g_\Delta^{\text{rest|1st}}$ increments this value by Δ. The REFs also update the label attribute indicating whether the first part of a rest must still be taken.

The REFs $g_\Delta^{\text{nightrest}}$ and $g_\Delta^{\text{nightrest|2nd}}$ update label attributes similar to REFs $g_\Delta^{\text{dayrest}}$ and $g_\Delta^{\text{dayrest|2nd}}$, however, they ensure that the rest spans over the entire night by setting

$$\hat{l}_\triangleleft^{\text{time}} := \min\{l_\triangleleft^{\text{time}} - \Delta, l_\triangleleft^{\text{dawn}} - t^{\text{night}}\}.$$

$\hat{l}_\triangleleft$	$g_\Delta^{\text{dayrest}}(l_\triangleleft)$	$g_\Delta^{\text{dayrest}\|\text{2nd}}(l_\triangleleft)$	$g_\Delta^{\text{rest}\|\text{1st}}(l_\triangleleft)$
$\hat{l}_\triangleleft^{\text{time}}$	$l_\triangleleft^{\text{time}} - \Delta$	$l_\triangleleft^{\text{time}} - \Delta$	$l_\triangleleft^{\text{time}} - \Delta$
$\hat{l}_\triangleleft^{\text{drive}\|\text{R}}$	0	0	
$\hat{l}_\triangleleft^{\text{drive}\|\text{B}}$	0	0	0
$\hat{l}_\triangleleft^{\text{work}\|\text{B}}$	0	0	0
$\hat{l}_\triangleleft^{\text{elapsed}\|\text{R}}$	t^{rest}	$t^{\text{rest}\|\text{2nd}}$	$l_\triangleleft^{\text{elapsed}\|\text{R}} + \Delta$
$\hat{l}_\triangleleft^{\text{earliest}\|\text{R}}$	$l_\triangleleft^{\text{dawn}} + t^{\text{rest}}$	$l_\triangleleft^{\text{dawn}} + t^{\text{rest}\|\text{1st}} + t^{\text{rest}\|\text{2nd}}$	$\max\{l_\triangleleft^{\text{earliest}\|\text{R}}, l_\triangleleft^{\text{dawn}} + l_\triangleleft^{\text{elapsed}\|\text{R}} + \Delta\}$
$\hat{l}_\triangleleft^{\text{rest}}$	0	$t^{\text{rest}\|\text{1st}}$	0

Table 7 Backward REFs related to rest periods taken during day time

Furthermore, they increment $l_\triangleleft^{\text{days}}$ by one, reduce $l_\triangleleft^{\text{dawn}}$ by t^{day}, and set $\hat{l}_\triangleleft^{\text{earliest}|\text{R}}$ to the appropriate value.

Table 8 shows how the REFs related to break periods update label attributes. These REFs

$\hat{l}_\triangleleft$	$g_\Delta^{\text{break}\|\text{2nd}}(l_\triangleleft)$	$g_\Delta^{\text{break}\|\text{1st}}(l_\triangleleft)$
$\hat{l}_\triangleleft^{\text{time}}$	$l_\triangleleft^{\text{time}} - \Delta$	$l_\triangleleft^{\text{time}} - \Delta$
$\hat{l}_\triangleleft^{\text{drive}\|\text{B}}$	0	
$\hat{l}_\triangleleft^{\text{work}\|\text{B}}$	0	
$\hat{l}_\triangleleft^{\text{elapsed}\|\text{R}}$	$l_\triangleleft^{\text{elapsed}\|\text{R}} + \Delta$	$l_\triangleleft^{\text{elapsed}\|\text{R}} + \Delta$
$l_\triangleleft^{\text{earliest}\|\text{R}}$	$\max\{l_\triangleleft^{\text{earliest}\|\text{R}}, l_\triangleleft^{\text{dawn}} + l_\triangleleft^{\text{elapsed}\|\text{R}} + l_\triangleleft^{\text{rest}} + \Delta + t^{\text{break}\|\text{1st}}\}$	$\max\{l^{\text{earliest}\|\text{R}}, l^{\text{dawn}} + l^{\text{elapsed}\|\text{R}} + l^{\text{rest}} + \Delta\}$
$\hat{l}_\triangleleft^{\text{break}}$	$t^{\text{break}\|\text{1st}}$	0

Table 8 Backward REFs related to break periods

reduce the time attribute by Δ, increase the time elapsed until the point in time where the rest is fully taken by Δ, and increase the earliest possible completion time of the next rest if the preceding activities could otherwise not be conducted after the end of the previous night. Furthermore, the REF $g_\Delta^{\text{break}|\text{2nd}}$ sets the amount of break required to the minimum duration of the first part of a break, and the REF $g_\Delta^{\text{break}|\text{1st}}$ sets the amount of break required to zero.

REF g_Δ^{idle} updates labels similar to REF $g_\Delta^{\text{break}|\text{1st}}$, except that it does not change $l_\triangleleft^{\text{break}}$.

In order to only consider labels complying with the regulations, the feasibility conditions given in Table 9 must be satisfied when using the corresponding REFs. For REF g_Δ^{drive} the duration Δ must not exceed the largest possible driving time given by

$$\begin{aligned}\Delta_{l_\triangleleft}^{\texttt{EU}} := \min\{&l_\triangleleft^{\text{trip}}, t^{\text{drive}|\text{W}} - l_\triangleleft^{\text{drive}|\text{W}}, t^{\text{drive}|\text{R}} - l_\triangleleft^{\text{drive}|\text{R}}, t^{\text{drive}|\text{B}} - l_\triangleleft^{\text{drive}|\text{B}},\\ &t^{\text{work}|\text{W}} - l_\triangleleft^{\text{work}|\text{W}}, t^{\text{work}|\text{B}} - l_\triangleleft^{\text{work}|\text{B}},\\ &t^{\text{day}} - l_\triangleleft^{\text{rest}} - l_\triangleleft^{\text{break}} - l_\triangleleft^{\text{elapsed}|\text{R}}, l_\triangleleft^{\text{time}} - l_\triangleleft^{\text{dawn}} - l_\triangleleft^{\text{rest}} - l_\triangleleft^{\text{break}}\}.\end{aligned} \tag{3}$$

REF	Feasibility conditions
$g_\Delta^{\text{drive}}(l_\triangleleft)$	$\Delta \leq \Delta^{\mathtt{EU}}_{l_\triangleleft}$
$g_n^{\text{visit}}(l_\triangleleft)$	$l_\triangleleft^{\text{trip}} = 0$, $l_\triangleleft^{\text{time}} \geq t_n^{\min} + s_n$, $l_\triangleleft^{\text{dawn}} \leq t_n^{\max}$, $s_n \leq t^{\text{work}\mid\text{B}} - l_\triangleleft^{\text{work}\mid\text{B}}$, $s_n \leq t^{\text{work}\mid\text{W}} - l_\triangleleft^{\text{work}\mid\text{W}}$, $\max\{l_\triangleleft^{\text{elapsed}} + s_n, l_\triangleleft^{\text{earliest}} - t_n^{\max}\} + l_\triangleleft^{\text{rest}} + l_\triangleleft^{\text{break}} \leq t^{\text{day}}$, $\min\{l_\triangleleft^{\text{time}} - s_n, t_n^{\max}\} - l_\triangleleft^{\text{rest}} - l_\triangleleft^{\text{break}} \geq l_\triangleleft^{\text{dawn}}$
$g_\Delta^{\text{dayrest}}(l_\triangleleft)$	$l_\triangleleft^{\text{rest}} = 0$, $l_\triangleleft^{\text{break}} = 0$, $\Delta \geq t^{\text{rest}}$, $\Delta \leq l_\triangleleft^{\text{time}} - l_\triangleleft^{\text{dawn}}$
$g_\Delta^{\text{dayrest}\mid\text{2nd}}(l_\triangleleft)$	$l_\triangleleft^{\text{rest}} = 0$, $l_\triangleleft^{\text{break}} = 0$, $\Delta \geq t^{\text{rest}\mid\text{2nd}}$, $\Delta \leq l_\triangleleft^{\text{time}} - l_\triangleleft^{\text{dawn}} - t^{\text{rest}\mid\text{1st}}$
$g_\Delta^{\text{nightrest}}(l_\triangleleft)$	$l_\triangleleft^{\text{rest}} = 0$, $l_\triangleleft^{\text{break}} = 0$, $\Delta \geq t^{\text{rest}}$, $\Delta \leq l_\triangleleft^{\text{time}} - l_\triangleleft^{\text{dawn}} + t^{\text{day}}$
$g_\Delta^{\text{nightrest}\mid\text{2nd}}(l_\triangleleft)$	$l_\triangleleft^{\text{rest}} = 0$, $l_\triangleleft^{\text{break}} = 0$, $\Delta \geq t^{\text{rest}\mid\text{2nd}}$, $\Delta \leq l_\triangleleft^{\text{time}} - l_\triangleleft^{\text{dawn}} - t^{\text{rest}\mid\text{1st}} + t^{\text{day}}$
$g_\Delta^{\text{rest}\mid\text{1st}}(l_\triangleleft)$	$l_\triangleleft^{\text{rest}} = t^{\text{rest}}$, $l_\triangleleft^{\text{break}} = 0$, $\Delta \geq t^{\text{rest}\mid\text{1st}}$, $\Delta \leq t^{\text{day}} - l_\triangleleft^{\text{elapsed}}$, $\Delta \leq l_\triangleleft^{\text{time}} - l_\triangleleft^{\text{dawn}}$
$g_\Delta^{\text{break}\mid\text{2nd}}(l_\triangleleft)$	$l_\triangleleft^{\text{break}} = 0$, $\Delta \geq t^{\text{break}\mid\text{2nd}}$, $\Delta \leq t^{\text{day}} - l_\triangleleft^{\text{elapsed}} - l_\triangleleft^{\text{rest}} - t^{\text{break}\mid\text{1st}}$, $\Delta \leq l_\triangleleft^{\text{time}} - l_\triangleleft^{\text{dawn}} - l_\triangleleft^{\text{rest}} - t^{\text{break}\mid\text{1st}}$
$g_\Delta^{\text{break}\mid\text{1st}}(l_\triangleleft)$	$l_\triangleleft^{\text{break}} = t^{\text{break}\mid\text{1st}}$, $\Delta \geq t^{\text{break}\mid\text{1st}}$, $\Delta \leq t^{\text{day}} - l_\triangleleft^{\text{elapsed}} - l_\triangleleft^{\text{rest}}$, $\Delta \leq l_\triangleleft^{\text{time}} - l_\triangleleft^{\text{dawn}} - l_\triangleleft^{\text{rest}}$
$g_\Delta^{\text{idle}}(l_\triangleleft)$	$\Delta \leq t^{\text{day}} - l_\triangleleft^{\text{elapsed}} - l_\triangleleft^{\text{rest}}$, $\Delta \leq l_\triangleleft^{\text{time}} - l_\triangleleft^{\text{dawn}} - l_\triangleleft^{\text{rest}}$

Table 9 Feasibility conditions in backward labeling

For REF g_n^{visit} the customer location must have been reached, the time window must not be closed, the time window must have opened after the previous night, limits on the cumulative working time must not be exceeded, it must be possible to take the rest within 24 hours after the previous rest, and the working period must not reach into the previous night. For REFs $g_\Delta^{\text{dayrest}}$, $g_\Delta^{\text{dayrest}\mid\text{2nd}}$, $g_\Delta^{\text{nightrest}}$, and $g_\Delta^{\text{nightrest}\mid\text{2nd}}$ the amount of break and rest required must be zero, Δ must be at least as large as required by the regulation, but small enough so that the rest does not reach into the previous night or the night before the previous night. For REFs $g_\Delta^{\text{rest}\mid\text{1st}}$, $g_\Delta^{\text{break}\mid\text{2nd}}$, and $g_\Delta^{\text{break}\mid\text{1st}}$, the duration Δ must be at least as large as required by the regulation, but small enough so that the respective activity begins before the next night and the rest can be completed within 24 hours after the previous rest. Furthermore, REF $g_\Delta^{\text{rest}\mid\text{1st}}$ requires that the second part of the rest is already taken and the first part of a break is not required, REF $g_\Delta^{\text{break}\mid\text{2nd}}$ requires that the first part of a break is not required, and $g_\Delta^{\text{break}\mid\text{1st}}$ requires that the second part of a break is already taken. Lastly, for REF g_Δ^{idle}, the duration Δ must be small enough so that the respective activity begins before the next night and the rest can be completed within 24 hours after the previous rest.

Dominance. Like in the forward labeling method we can use dominance rules to reduce the number of alternative labels to be considered. Given two feasible labels $l_\triangleleft$ and $\bar{l}_\triangleleft$ which both represent a driver state at the beginning of the partial route $(n_i, n_{i+1}, \ldots, n_k)$ with $1 \leq i \leq k$, we write $l_\triangleleft \preceq \bar{l}_\triangleleft$ if $l_\triangleleft^j \leq \bar{l}_\triangleleft^j$ for all $j \in \{$days, trip, drive|W, drive|R, drive|B, work|W, work|B, elapsed, earliest, rest, break$\}$ and $l_\triangleleft^{\text{time}} \geq \bar{l}_\triangleleft^{\text{time}}$. Note that if $l_\triangleleft^{\text{time}} \geq \bar{l}_\triangleleft^{\text{time}}$ and $l_\triangleleft^{\text{earliest}} \leq \bar{l}_\triangleleft^{\text{earliest}}$ then $l_\triangleleft^{\text{dawn}} = \bar{l}_\triangleleft^{\text{dawn}}$.

If $l_\triangleleft \preceq \bar{l}_\triangleleft$, then we also have $g(l_\triangleleft) \preceq g(\bar{l}_\triangleleft)$ for each REF $g \in \{g^{\text{trip}}_{nm}, g^{\text{drive}}_\Delta, g^{\text{dayrest}}_\Delta, g^{\text{dayrest|2nd}}_\Delta, g^{\text{nightrest}}_\Delta, g^{\text{nightrest|2nd}}_\Delta, g^{\text{rest|1st}}_\Delta, g^{\text{break|2nd}}_\Delta, g^{\text{break|1st}}_\Delta, g^{\text{visit}}_n\}$ because all REFs are non-decreasing in all resources. Hence, $l_\triangleleft$ dominates $\bar{l}_\triangleleft$ and $\bar{l}_\triangleleft$ can be discarded from the set of labels to be updated.

In the Appendix it is shown which sequences of driver activities are dominated by others, why it is always better to schedule driving activities as long as possible, and why it is always better to schedule break and rest periods as short as possible.

REF	Inferiority conditions
$g^{\text{idle}}_\Delta(l_\triangleleft)$	always
$g^{\text{drive}}_\Delta(l_\triangleleft)$	$\Delta < \Delta^{\mathtt{EU}}_{l_\triangleleft}$
$g^{\text{dayrest}}_\Delta(l_\triangleleft)$	$\Delta^{\mathtt{EU}}_{l_\triangleleft} > 0$ or $\Delta > t^{\text{rest}}$ or $l^{\text{last}}_\triangleleft \in \{g^{\text{dayrest}}, g^{\text{nightrest}},\}$ or (g^{visit}_n is feasible and $l^{\text{time}}_\triangleleft \leq t^{\max}_n + s_n$)
$g^{\text{dayrest\|2nd}}_\Delta(l_\triangleleft)$	$\Delta^{\mathtt{EU}}_{l_\triangleleft} > 0$ or $\Delta > t^{\text{rest\|2nd}}$ or $l^{\text{last}}_\triangleleft \in \{g^{\text{dayrest}}, g^{\text{nightrest}}\}$ or (g^{visit}_n is feasible and $l^{\text{time}}_\triangleleft \leq t^{\max}_n + s_n$)
$g^{\text{nightrest}}_\Delta(l_\triangleleft)$	$\Delta^{\mathtt{EU}}_{l_\triangleleft} > 0$ or $\Delta > t^{\text{rest}}$ or $l^{\text{last}}_\triangleleft \in \{g^{\text{dayrest}}\}$ or (g^{visit}_n is feasible and $l^{\text{time}}_\triangleleft \leq t^{\max}_n + s_n$)
$g^{\text{nightrest\|2nd}}_\Delta(l_\triangleleft)$	$\Delta^{\mathtt{EU}}_{l_\triangleleft} > 0$ or $\Delta > t^{\text{rest\|2nd}}$ or $l^{\text{last}}_\triangleleft \in \{g^{\text{dayrest}}\}$ or (g^{visit}_n is feasible and $l^{\text{time}}_\triangleleft \leq t^{\max}_n + s_n$)
$g^{\text{rest\|1st}}_\Delta(l_\triangleleft)$	$\Delta^{\mathtt{EU}}_{l_\triangleleft} > 0$ or $\Delta > t^{\text{rest\|1st}}$ or $l^{\text{last}}_\triangleleft \in \{g^{\text{dayrest\|2nd}}, g^{\text{nightrest\|2nd}}\}$ or (g^{visit}_n is feasible and $l^{\text{time}}_\triangleleft \leq t^{\max}_n + s_n$)
$g^{\text{break\|2nd}}_\Delta(l_\triangleleft)$	$\Delta^{\mathtt{EU}}_{l_\triangleleft} > 0$ or $\Delta > t^{\text{break\|2nd}}$ or $l^{\text{work\|B}}_\triangleleft = 0$ or (g^{visit}_n is feasible and $l^{\text{time}}_\triangleleft \leq t^{\max}_n + s_n$)
$g^{\text{break\|1st}}_\Delta(l_\triangleleft)$	$\Delta^{\mathtt{EU}}_{l_\triangleleft} > 0$ or $\Delta > t^{\text{break\|1st}}$ or (g^{visit}_n is feasible and $l^{\text{time}}_\triangleleft \leq t^{\max}_n + s_n$)

Table 10 Inferiority conditions in backward labeling

The conditions in Table 10 show when a label generated by a backward REF is dominated by another label. In particular, they tell us that it is never beneficial to explicitly schedule idle periods, that it is always better to schedule driving activities as long as possible, and that it is always better to schedule break and rest periods as short as possible. Also a break or rest should not be scheduled if the last activity was a break or rest or if it is possible to schedule the next visit.

Combining forward and backward labels. A bidirectional labeling method for EU hours of service regulations can be obtained by determining forward labels for a partial route $(n_1, n_2, \ldots, n_i)$, backward labels for a partial route $(n_i, n_{i+1}, \ldots, n_k)$, and checking the conditions for a feasible merge for each pair of forward and backward label. Following Goel (2018), a forward label can be represented by

$$l_\triangleright = (l^{\text{time}}_\triangleright, l^{\text{trip}}_\triangleright, l^{\text{work|W}}_\triangleright, l^{\text{drive|W}}_\triangleright, l^{\text{drive|R}}_\triangleright, l^{\text{work|B}}_\triangleright, l^{\text{drive|B}}_\triangleright, l^{\text{elapsed|R}}_\triangleright, l^{\text{latest|R}}_\triangleright, l^{\text{rest}}_\triangleright, l^{\text{break}}_\triangleright, l^{\text{days}}_\triangleright, l^{\text{dusk}}_\triangleright)$$

where $l_{\triangleright}^{\text{time}}$, $l_{\triangleright}^{\text{trip}}$, $l_{\triangleright}^{\text{work|W}}$, $l_{\triangleright}^{\text{drive|W}}$, $l_{\triangleright}^{\text{drive|R}}$, $l_{\triangleright}^{\text{work|B}}$, $l_{\triangleright}^{\text{drive|B}}$, and $l_{\triangleright}^{\text{days}}$ are the counterparts of the respective backward labels and

$l_{\triangleright}^{\text{elapsed|R}}$ represents the time elapsed since the end of the previous rest,

$l_{\triangleright}^{\text{latest|R}}$ represents the latest possible time at which the previous rest must be completed,

$l_{\triangleright}^{\text{rest}}$ represents the remaining amount of rest required, i.e., t^{rest} or $t^{\text{rest|2nd}}$,

$l_{\triangleright}^{\text{break}}$ represents the remaining amount of break time required, i.e., t^{break} or $t^{\text{break|2nd}}$, and

$l_{\triangleright}^{\text{dusk}}$ represents the time at which the next night begins.

We now show how a forward label $l_{\triangleright}$ associated to a driver state upon completion of a partial route $(n_1, n_2, \ldots, n_i)$ can be combined with a backward label $l_{\triangleleft}$ associated to a driver state when beginning a partial route $(n_i, n_{i+1}, \ldots, n_k)$. Recall that both the forward and the backward labeling method add the stationary work at location n_i and the respective duration s_{n_i} must not be double counted when combining the labels.

A necessary condition for combining the forward label with the backward label obviously is that

$$l_{\triangleright}^{\text{work|W}} + l_{\triangleleft}^{\text{work|W}} - s_{n_i} \leq t^{\text{work|W}} \tag{4a}$$

$$l_{\triangleright}^{\text{drive|W}} + l_{\triangleleft}^{\text{drive|W}} \leq t^{\text{drive|W}} \tag{4b}$$

because otherwise the cumulative amounts of driving and work would exceed the weekly limits. If furthermore

$$\max\{l_{\triangleright}^{\text{time}} + l_{\triangleright}^{\text{rest}}, l_{\triangleright}^{\text{dusk}} + t^{\text{night}}\} \leq l_{\triangleleft}^{\text{time}} + s_{n_i} - l_{\triangleleft}^{\text{rest}} - l_{\triangleleft}^{\text{break}}, \tag{5}$$

both labels can be merged because at least one night rest can be scheduled between the respective partial schedules excluding the work of the backward label. The total number of days required for the schedule corresponding to the merged pair of labels is

$$l_{\triangleright}^{\text{days}} + l_{\triangleleft}^{\text{days}} + \max\left\{0, \left\lfloor \frac{l_{\triangleleft}^{\text{dawn}} - l_{\triangleright}^{\text{dusk}}}{t^{\text{day}}} \right\rfloor\right\},$$

i.e., the sum of the duration of both partial schedules plus the number of full days in between.

If no night rest can be scheduled, both labels can be merged if

$$l_{\triangleright}^{\text{time}} - s_{n_i} \leq l_{\triangleleft}^{\text{time}} \tag{6a}$$

$$l_{\triangleright}^{\text{dusk}} \geq l_{\triangleleft}^{\text{dawn}} \tag{6b}$$

$$l_{\triangleright}^{\text{elapsed}} + l_{\triangleleft}^{\text{elapsed}} - s_{n_i} \leq t^{\text{day}} \tag{6c}$$

$$l_{\triangleleft}^{\text{earliest}} - l_{\triangleright}^{\text{latest}} \leq t^{\text{day}} \tag{6d}$$

$$l_{\triangleright}^{\text{drive|R}} + l_{\triangleleft}^{\text{drive|R}} \leq t^{\text{drive|R}} \tag{6e}$$

$$l_{\triangleright}^{\text{drive|B}} + l_{\triangleleft}^{\text{drive|B}} \leq t^{\text{drive|B}} \tag{6f}$$

$$l_{\triangleright}^{\text{work|B}} + l_{\triangleleft}^{\text{work|B}} - s_{n_i} \leq t^{\text{work|B}} \tag{6g}$$

$$(l_{\triangleright}^{\text{rest}} = t^{\text{rest|2nd}}) = (l_{\triangleleft}^{\text{rest}} = t^{\text{rest|1st}}) \tag{6h}$$

$$(l_{\triangleright}^{\text{break}} = t^{\text{break|2nd}}) = (l_{\triangleleft}^{\text{break}} = t^{\text{break|1st}}). \tag{6i}$$

In this case, the total number of days required for the schedule corresponding to the merged pair of labels is $l_{\triangleright}^{\text{days}} + l_{\triangleleft}^{\text{days}} - 1$.

Analogously, if above conditions hold for $f_{t^{\text{break|1st}}}^{\text{break|1st}}(l_{\triangleright})$, $f_{l_{\triangleright}^{\text{break}}}^{\text{break}}(l_{\triangleright})$, $f_{t^{\text{rest|1st}}}^{\text{rest|1st}}(l_{\triangleright})$, or $f_{l_{\triangleright}^{\text{rest}}}^{\text{dayrest}}(l_{\triangleright})$, then it is possible to merge the backward label with the forward label obtained by adding the required break or rest period.

5. Vehicle Routing

This section describes a BPC algorithm for solving the VRTDSP using the bidirectional labeling approaches presented in the previous section.

Let C denote a given set of customer locations. For each $n \in C$ let $[t_n^{\min}, t_n^{\max}]$, s_n, and q_n denote the time window of the customer, the non-negative duration of the service time that must begin within the time window, and the non-negative demand. Furthermore, let n^{depot} denote the depot at which a homogeneous fleet of K vehicles are located, each having a capacity of Q. Analogously to customer locations, the depot has an associated time window, service time, and demand, however the time window spans the entire planning horizon, and the service time and demand are zero. For each pair $(n,m) \in C \cup \{n^{\text{depot}}\} \times C \cup \{n^{\text{depot}}\}$, let d_{nm} and c_{nm} denote the driving time (excluding break and rest times) and the distance-related costs of travelling between n and m. The VRTDSP calls for the determination of at most K routes, where a route $r = (n_1, n_2, \ldots, n_k)$ is feasible if it starts and ends at the depot, i.e., $n_1 = n_k = n^{\text{depot}}$, if it visits a subset of customer locations between start and end, i.e., $n_i \in C$ for $1 < i < k$, if the capacity is not exceeded, i.e., $\sum_{i=1}^{k} q_{n_i} \leq Q$, and if a feasible truck driver schedule exists for the route. The goal is to find a set of feasible routes such that each customer in C is visited by exactly one route and that the total costs for all routes are minimized. As mentioned previously, labor costs in the United States are usually based on distance travelled and, therefore, we determine the costs of a route $r = (n_1, n_2, \ldots, n_k)$ by

$$c_r = \sum_{i=1}^{k-1} c_{n_i n_{i+1}}. \tag{7}$$

In the European Union labour cost must not be based on distance travelled and therefore we assume the costs of a route $r = (n_1, n_2, \ldots, n_k)$ to be the weighted sum of distance-related costs, in particular, for fuel and toll, and duration-related costs, in particular, for daily driver wages. These costs can be determined by

$$c_r = \sum_{i=1}^{k-1} c_{n_i n_{i+1}} + c^{\text{day}} \cdot \min_{l \in L_r} \{l^{\text{days}}\} \tag{8}$$

where L_r denotes the set of labels corresponding to feasible truck driver schedules for route r and c^{day} denotes the cost for each day of operation.

The VRTDSP can be formulated as

$$\min \quad \sum_{r \in R} c_r \lambda_r \tag{9a}$$

$$\text{s.t.} \quad \sum_{r \in R} a_{nr} \lambda_r = 1 \qquad \forall n \in C \tag{9b}$$

$$\sum_{r \in R} \lambda_r \leq K \tag{9c}$$

$$\lambda_r \in \{0, 1\} \qquad \forall r \in R \tag{9d}$$

In this formulation R denotes the set of all feasible routes. The binary parameter a_{nr} indicates whether customer n is visited by route r, and the binary variable λ_r indicates if route r is used in the solution. The objective function (9a) is to minimize the cumulative cost over all routes used in the solution. Constraints (9b) ensure that each customer is visited exactly once. The number of used vehicles is limited by (9c) and the variable domains are given in (9d).

This formulation suffers from the usually huge number of routes in the set R. To overcome this issue we use a CG algorithm algorithm to solve the problem. Therein, the set R is replaced by a small subset $\bar{R} \subset R$ of routes and more routes are added dynamically to $\bar{R}$ until a solution of the overall problem is found. The linear relaxation of Formulation (9) in which R is replaced by $\bar{R}$ is called the *restricted master program* (RMP). The CG algorithm alternates between optimizing the linear relaxation of the RMP and solving a pricing problem that adds additional variables to the RMP. The pricing problem asks for a route r with negative reduced cost $\bar{c}_r := c_r - \mu - \sum_{n \in C} a_{nr} \pi_n < 0$ where π_n denote the dual prices of the constraints (9b) and μ denotes the dual price of the convexity constraint (9c) associated with the current solution.

Routes with negative reduced costs can be found by solving a SPPRC where the distance-related arc costs c_{nm} are replaced by the reduced arc cost $\bar{c}_{nm} := c_{nm} - \frac{1}{2}\pi_n - \frac{1}{2}\pi_m$ with $\pi_{n^{\text{depot}}} = \mu$. The reduced costs of a route $r = (n_1, n_2, \ldots, n_k)$ in the United States can be determined by

$$\bar{c}_r = \sum_{i=1}^{k-1} \bar{c}_{n_i n_{i+1}} \tag{10}$$

and in the European Union by

$$\bar{c}_r = \sum_{i=1}^{k-1} \bar{c}_{n_i n_{i+1}} + c^{\text{day}} \cdot \min_{l \in L_r} \{l^{\text{days}}\}. \tag{11}$$

If no route with negative reduced costs can be found, the solution is optimal for the linear relaxation of the original problem. Otherwise, the route with negative reduced costs is added to $\bar{R}$ and the RMP is solved again. After an optimal solution for the linear relaxation of the original problem is found, branch-and-bound may be required to find an optimal integer solution of the original problem.

5.1. Shortest path problem with resource constraints

The pricing problem is a SPPRC that can be solved using labeling methods based on the methods presented in Section 4. Herein, forward labels are expanded by additional attributes $l_\triangleright^{\text{visited}}$, $l_\triangleright^{\text{cost}}$, and $l_\triangleright^{\text{load}}$ and backward labels by additional attributes $l_\triangleleft^{\text{visited}}$, $l_\triangleleft^{\text{cost}}$, and $l_\triangleleft^{\text{load}}$, where

$l_\triangleright^{\text{visited}}, l_\triangleleft^{\text{visited}}$ represent the set of customer locations already visited,
$l_\triangleright^{\text{cost}}, l_\triangleleft^{\text{cost}}$ represent the (reduced) cost of the partial route, and
$l_\triangleright^{\text{load}}, l_\triangleleft^{\text{load}}$ represent the cumulated demand of the visited customers.

The REFs f_{nm}^{trip} and g_{nm}^{trip} are changed so that they increase $l_\triangleright^{\text{cost}}$ and $l_\triangleleft^{\text{cost}}$ by $\bar{c}_{nm}$. For EU hours of service regulations, REFs $f^{\text{nightrest}}$, $g^{\text{nightrest}}$, and $g^{\text{nightrest}|2\text{nd}}$ are changed so that they increase $l_\triangleright^{\text{cost}}$ and $l_\triangleleft^{\text{cost}}$ by c^{day}. Furthermore, REFs f_m^{visit} and g_n^{visit} update $l_\triangleright^{\text{visited}}$, $l_\triangleleft^{\text{visited}}$, $l_\triangleright^{\text{load}}$, and $l_\triangleleft^{\text{load}}$ accordingly. The conditions that the next customer is not yet visited and that the capacity of the vehicle is not exceeded are added to the feasibility conditions of REFs f_{nm}^{trip} and g_{nm}^{trip} to avoid unnecessary calculations. Furthermore, the dominance criteria are extended by the conditions that a forward label $l_\triangleright$ can only dominate another forward label $\bar{l}_\triangleright$ if $l_\triangleright^{\text{visited}} \subseteq \bar{l}_\triangleright^{\text{visited}}$, $l_\triangleright^{\text{cost}} \leq \bar{l}_\triangleright^{\text{cost}}$, and $l_\triangleright^{\text{load}} \leq \bar{l}_\triangleright^{\text{load}}$. Analogously the dominance criteria are extended for backward labels. Note that for EU hours of service regulations, we remove the conditions that $l_\triangleright^{\text{days}} \leq \bar{l}_\triangleright^{\text{days}}$ and $l_\triangleleft^{\text{days}} \leq \bar{l}_\triangleleft^{\text{days}}$ because the objective is to minimize the weighted sum of distance- and duration-related costs which is represented by $l_\triangleright^{\text{cost}}$ and $l_\triangleleft^{\text{cost}}$.

Bidirectional labeling is used to solve the SPPRC using a *dynamic half-way point* defined on the resources $l_\triangleright^{\text{time}}$ and $l_\triangleleft^{\text{time}}$. As proposed by Tilk et al. (2017), the bidirectional labeling iteratively selects a forward or a backward label to extend, and dynamically computes a half-way point. Forward labels with a value of $l_\triangleright^{\text{time}}$ larger than this half-way point, and backward labels with a value of $l_\triangleleft^{\text{time}}$ smaller than this half-way point, are not extended and the method terminates when no label remains to be extended. After termination of the method, forward and backward labels are merged. When merging a forward label $l_\triangleright$ for a partial route $(n_1, n_2, \dots, n_i)$ and a backward label $l_\triangleleft$ for a partial route $(n_i, n_{i+1}, \dots, n_k)$, the conditions

$$|l_\triangleright^{\text{visited}} \cap l_\triangleleft^{\text{visited}}| = 1 \tag{12a}$$

$$l_\triangleright^{\text{load}} + l_\triangleleft^{\text{load}} - q_{n_i} \leq Q \tag{12b}$$

as well as the other merge conditions presented in Section 4 must hold.

The labeling takes by far the largest portion of the computation time in the overall BPC algorithm. To speed up the solution process, four acceleration techniques are used. First, we use the *ng-path relaxation* (Baldacci et al. 2011) with a neighborhood size of ten. Second, an additional set of *unreachable customers* is defined to strengthen the dominance as proposed by Feillet et al. (2004). Third, the labeling is solved heuristically using a *limited discrepancy search* (LDS, see Feillet et al. 2007). Last, a heuristic dominance rule is applied to further strengthen the dominance.

5.2. Branching and Cutting

To strengthen the linear relaxation, two classes of valid inequalities are used:. First, *2-path inequalities* (Kohl et al. 1999) are separated and added whenever they are violated. Let $W \subset C$ be a subset of customers that can not be visited by one single vehicle due to capacity or time window restrictions. Moreover, let $\delta^-(W)$ be the set of all arcs $(i,j) \in A$ with $i \in W$ and $j \notin W$. The corresponding 2-path inequality is given by $\sum_{r\in R}\sum_{(i,j)\in\delta^-(W)} b^r_{ij}\lambda_r \geq 2$, where b^r_{ij} is the number of times route r traverses arc $(i,j) \in A$. We use the heuristic proposed by Kohl et al. (1999) to generate candidate sets W of maximal cardinality.

Second, *subset-row inequalities* (Jepsen et al. 2008) defined on sets of three customer are separated at the root node of the branch-and-bound tree. The inequality for a customer set $U_k \subset C$ is given by $\sum_{r\in R}\lfloor\frac{h^r}{2}\rfloor\lambda_r \leq 1$, where h^r is the number of times route r visits a customer in U_k. Note that the use of subset-row inequalities in the master problem requires adjustments in the pricing problem as explained by Jepsen et al. (2008).

Branching on arcs is required to finally ensure integer solutions of Formulation 9. To accelerate the solution process, we apply *strong branching* with up to eight candidate arcs. We choose the eight most fractional arcs in the current solution as branching candidates and perform a rough evaluation of each candidate by solving the current RMP twice, adding the constraint corresponding to each child node without generating additional columns. A similar procedure was applied for the capacitated VRP by Pecin et al. (2016). The resulting improvements in the lower bounds are usually overestimated. However, this evaluation strategy is fast and beneficial compared to just choosing the most fractional arc too branch on. The arc to branch on is then chosen according to the product rule (Achterberg 2007). As branch-and-bound node-selection rule, we apply a best-bound-first strategy, because our primary goal is to improve the dual bound.

6. Computational Results

This section reports on computational experiments conducted to evaluate the bidirectional approach and the overall performance of our BPC algorithm. We implemented our algorithm in C++ and compiled it into 64-bit single-thread code with MS Visual Studio 2013. All experiments were conducted on a standard PC with an Intel(R) Core(TM)i7-5930k clocked at 3.5 GHz and 64 GB of RAM, by allowing a single thread for each run. CPLEX 12.6.2 was used with the default parameters to solve the RMP in the column-generation algorithm and to determine an integer solution based on the columns generated so far when reaching the time limit of two hours. The time allowed to find the best integer solution in this final step was restricted to at most 600 seconds.

For US regulations, we tested our algorithm on the 56 benchmark instances for the VRTDSP-US proposed by Goel (2009) which can be obtained at `https://www.telematique.eu/research/downloads`. These instances are derived from the VRPTW benchmark instances of Solomon (1987) that can be grouped in six different classes: Randomly distributed customers (R1 and R2), clustered customers (C1 and C2) and a mixed distribution (RC1 and RC2). Instance classes R1, C1 and RC1 have tight time windows and strict vehicle capacity, while C2, RC2, and R2 have wide time windows and loose vehicle capacity. Each instance contains 100 customers, the service time at every customer is set to 60 minutes, and we assume an average speed of 70 km/h and a cost structure of 0.50 Euro per kilometer. Like Goel and Irnich (2017), we create smaller instances by considering only the first 25 or 50 customers. For the EU regulations, we use the same instance set as for US regulations but adapt the time windows as described in Goel (2018) such that every customer can be visited during day time. For our experiments, we considered different night time definitions from 20.00h to 7.00h, from 23.00h to 6.00h, and from 0.00h to 4.00h. These night time definitions are representative for a large share of the countries in the European Union. Moreover, we assume a daily cost of $c^{\text{day}} = 150$ Euro.

Table 11 and 12 contain aggregated results of our experiments for the linear relaxation (LP) and the integer program (IP) and compares them with results for the branch-and-price (BP) algorithms presented by Goel and Irnich (2017) and Goel (2018), respectively. The table reports results for the LP and IP, showing the number of instances for which an optimal solution was found and the average time (in seconds) required. For instances that are not solved to optimality the run time limit was used when calculating the average computation time. Detailed results for all instances are provided in the Appendix.

	State-of-the-art[1]				Our Approach[2]			
	LP		IP		LP		IP	
|C|	#Solved	Time[s]	#Solved	Time[s]	#Solved	Time[s]	#Solved	Time[s]
25	56	36.73	55	276.09	56	45.29	56	182.69
50	45	1698.05	30	3764.41	51	1264.69	36	3247.10

[1] CPU: Intel i7-5600U, run time limit: 7200 seconds
[2] CPU: Intel i7-5930k, run time limit: 7200 seconds

Table 11 Results for US regulation minimizing distance costs

Table 11 shows our results for US regulations compared to the unidirectional approach of Goel and Irnich (2017). Regarding the 25-customer instances, we can solve the last remaining instance RC208 to optimality. For the 50-customer instances, we can solve six more instances to proven optimality and the runtime decreases by around 15 %.

		State-of-the-art[1] IP		Our Approach[2] LP		Our Approach[2] IP	
$\|C\|$	Night Time	#Solved	Time[s]	#Solved	Time[s]	#Solved	Time[s]
25	20h – 7h	7	3313.90	56	33.11	53	475.03
25	23h – 6h	6	3289.81	55	355.14	50	1224.35
25	0h – 4h	5	3346.11	55	590.35	49	1377.25
50	20h – 7h	N/A		51	1238.14	37	3069.85
50	23h – 6h	N/A		43	2726.42	24	4505.31
50	0h – 4h	N/A		37	3632.84	21	4965.32

[1] CPU: Intel i7-5600U, run time limit: 3600 seconds
[2] CPU: Intel i7-5930k, run time limit: 7200 seconds

Table 12 Results for the EU regulation minimizing costs based on distance and duration

Regarding EU regulations, Table 12 shows that 152 out of 168 instances with 25 customers can be solved to optimality in around 1025 seconds on average. The average gap over the remaining instance for which the linear relaxation is solved is around three percent for the 25 customer instances. The linear relaxation is solved for almost all 25 customer instances in less than 330 seconds on average. Furthermore, our approach was able to solve almost half of the 50 customer instances to optimality within the run time limit. The average gap over the remaining 50 customer instance for which the linear relaxation was solved is around 4.5% on average. The share of 50 customer instances for which the linear relaxation is not solved within two hours is around 20% of the instances. Although our experiments used a run time limit of two hours instead of the one hour time limit used by Goel (2018), we can see that our approach clearly outperforms the BP approach with unidirectional labeling.

In order to better understand the contribution of the bidirectional labeling proposed in this paper compared to the other algorithmic differences, we ran the same experiments replacing bidirectional labeling in our BPC with pure forward labeling and pure backward labeling. Tables 13 and 14 show aggregated results comparing the results of our BPC algorithm where the subproblem is either solved with forward, backward, or bidirectional labeling. The tables contain the number of instances solved to optimality within the run time limit and the average runtime (in seconds) which is computed only over those instances that are solved to optimality by all three variants. It must be noted that the times reported for the bidirectional variant are significantly smaller compared to the values shown in Tables 11 and 12, because some of the very time-consuming instances are excluded from the average as they are not solved by the unidirectional variants.

	Forward Labeling		Backward Labeling		Bidirectional Labeling	
$\|C\|$	#Solved	Time[s]	#Solved	Time[s]	#Solved	Time[s]
25	55	199.31	55	298.42	56	125.62
50	30	1192.98	31	1149.73	36	483.23

Table 13 Comparison of uni- and bidirectional Labeling for US regulations

Table 13 shows that for US regulations, bidirectional labeling is on average between 1.5 times and 2.5 time faster than the unidirectional variants. Moreover, bidirectional labeling allows to solve six more instances to proven optimality. While backward labeling appears to be slower than forward labeling for the 25-customer instances, one more 50-customer instance can be solved to proven optimality with backward labeling.

Regarding EU regulations, an interesting observation is that backward labeling performs much worse than forward labeling. Table 14 shows that forward labeling solves 23 more instances and the computational effort is significantly lower. We ascribe this to the additional labels needed for schedules terminating with a full rest or the second part of a rest. Despite the comparably low performance of backward labeling, we can see that bidirectional labeling clearly outperforms unidirectional labeling. With bidirectional labeling, 28 more instances can be solved to optimality compared to forward labeling and 51 more instances compared to backward labeling. Bidirectional labeling is on average between 2 and 10 times faster than forward labeling and between 4 and 16 times faster than backward labeling. This shows that the bidirectional labeling method has a significant contribution to the good performance of our algorithm, especially considering EU regulations. One reason for this good performance is that in the EU-VRTDSP, initial labels have to be generated for each day of the planning horizon. In unidirectional labeling all of these alternative labels must be extended, which leads to a significantly higher computational burden. In our birectional approach many of the initial labels are never extended because they are already behind the half-way point.

7. Conclusion

In this paper we propose backward labeling methods for truck driver scheduling in the United States and the European Union. We show how labels generated with a forward labeling method can be combined with labels generated with our backward method. Being able to combine forward and backward labels can significantly speed up heuristic solution approaches for vehicle routing and truck driver scheduling problems, because unnecessary computational effort can be avoided when

		Forward Labeling		Backward Labeling		Bidirectional Labeling	
$\lvert C \rvert$	Night Time	#Solved	Time[s]	#Solved	Time[s]	#Solved	Time[s]
25	20h – 7h	51	409.81	46	673.10	53	41.74
25	23h – 6h	46	439.40	42	1114.34	50	131.55
25	0h – 4h	45	365.20	41	892.03	49	185.66
50	20h – 7h	23	340.59	23	725.61	37	100.78
50	23h – 6h	22	515.87	17	1112.53	24	261.92
50	0h – 4h	19	712.28	14	1348.48	21	250.62

CPU: Intel i7-5930k, run time limit: 7200 seconds

Table 14 Comparison of uni- and bidirectional Labeling for EU regulations

evaluating local changes to existing routes. Therefore, we expect bidirectional labeling to become a standard component in all local search based heuristics for the VRTDSP.

We present an exact branch-and-price-and-cut algorithm for solving the VRTDSP using a bidirectional labeling approach and show that our algorithm clearly outperforms uni-directional approaches. For US regulations, we can solve all 25 customer instances with an average computation time clearly below 5 minutes. For EU regulations, we can solve 152 out of 168 instances with 25 customers with an average computation time of around 17 minutes.

An important contribution of our approach is that it is particularly well suited for problems in which schedule durations must be minimized. As EU regulations prohibit any payment related to travel distance, labor costs cannot be included in the mileage costs. For realistic cost functions based on distance and duration, our bidirectional approach for the VRTDSP-EU is on average between 2 and 16 times faster than unidirectional variants.

References

Achterberg, T. 2007. Constraint integer programming. Ph.D. dissertation, Technische Universität Berlin. doi:10.14279/depositonce-1634.

Archetti, C., M. W. P. Savelsbergh. 2009. The trip scheduling problem. *Transportation Science* **43**(4) 417–431. doi:10.1287/trsc.1090.0278.

Baldacci, R., A. Mingozzi, R. Roberti. 2011. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research* **59**(5) 1269–1283. doi:10.1287/opre.1110.0975.

Bureau of Labor Statistics, U.S. Department of Labor. 2019. Occupational Outlook Handbook, Heavy and Tractor-trailer Truck Drivers. Available online at `https://www.bls.gov/ooh/transportation-and-material-moving/heavy-and-tractor-trailer-truck-drivers.htm` (visited March 09, 2019).

Campbell, A.M., M. Savelsbergh. 2004. Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science* **38**(3) 369–378.

Ceselli, A., G. Righini, M. Salani. 2009. A column generation algorithm for a rich vehicle-routing problem. *Transportation Science* **43**(1) 56–69. doi:10.1287/trsc.1080.0256.

Desaulniers, G., J. Desrosiers, M.M. Solomon, eds. 2005. *Column Generation*. Springer, New York, NY.

European Union. 2002. 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. *Official Journal of the European Communities* **L 80** 35–39.

European Union. 2006. 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. *Official Journal of the European Union* **L 102** 1–13.

Federal Motor Carrier Safety Administration. 2011. Hours of service of drivers. Federal Register Vol. 76, No. 248, p. 81134 - 81188.

Feillet, D., P. Dejax, M. Gendreau, C. Guéguen. 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks* **44**(3) 216–229.

Feillet, D., M. Gendreau, L.-M. Rousseau. 2007. New refinements for the solution of vehicle routing problems with branch and price. *INFOR* **45**(4) 239–256.

Goel, A. 2009. Vehicle scheduling and routing with drivers' working hours. *Transportation Science* **43**(1) 17–26. doi:10.1287/trsc.1070.0226.

Goel, A. 2010. Truck driver scheduling in the European Union. *Transportation Science* **44**(4) 429–441. doi:10.1287/trsc.1100.0330.

Goel, A. 2012. The minimum duration truck driver scheduling problem. *EURO Journal on Transportation and Logistics* **1**(4) 285–306. doi:10.1007/s13676-012-0014-9.

Goel, A. 2014. Hours of service regulations in the United States and the 2013 rule change. *Transport Policy* **33** 48–55.

Goel, A. 2018. Legal aspects in road transport optimization in Europe. *Transportation Research Part E: Logistics and Transportation Review* **114** 144–162. doi:10.1016/j.tre.2018.02.011.

Goel, A., S. Irnich. 2017. An exact method for vehicle routing and truck driver scheduling problems. *Transportation Science* **51**(2) 737–754.

Goel, A., L. Kok. 2012. Truck driver scheduling in the United States. *Transportation Science* **46**(3) 317–326. doi:10.1287/trsc.1110.0382.

Goel, A., T. Vidal. 2014. Hours of service regulations in road freight transport: An optimization-based international assessment. *Transportation Science* **48**(3) 391–412. doi:10.1287/trsc.2013.0477.

Goudswaard, A., B. Kuipers, N. Schoenmaker, I.L.D. Houtman, K. Jettinghof, P.A.J. Ruijs, W.M. Savenije, D.S.C. Osinga, M. Koomen. 2006. Road Transport Working Time Directive. Self-employed and Night Time Provisions. TNO Report R0622373/018-31364, DG Energy and Transport.

Irnich, S. 2008. Resource extension functions: Properties, inversion, and generalization to segments. *OR Spectrum* **30**(1) 113–148. doi:10.1007/s00291-007-0083-6.

Irnich, S., G. Desaulniers. 2005. Shortest path problems with resource constraints. Desaulniers et al. (2005), chap. 2, 33–65.

Jepsen, M., B. Petersen, S. Spoorendonk, D. Pisinger. 2008. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research* **56**(2) 497–511. doi:10.1287/opre.1070.0449.

Kohl, N., J. Desrosiers, O.B.G. Madsen, M.M. Solomon, F. Soumis. 1999. 2-path cuts for the vehicle routing problem with time windows. *Transportation Science* **33**(1) 101–116.

Kok, L., C. M. Meyer, H. Kopfer, J. M. 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. doi:10.1287/trsc.1100.0331.

Pecin, D., A. Pessoa, M. Poggi, E. Uchoa. 2016. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation* 1–40doi:10.1007/s12532-016-0108-8.

Prescott-Gagnon, E., G. Desaulniers, M. Drexl, L. M. Rousseau. 2010. European driver rules in vehicle routing with time windows. *Transportation Science* **44**(4) 455–473. doi:10.1287/trsc.1100.0328.

Rancourt, M. E., J. F. Cordeau, G. Laporte. 2013. Long-haul vehicle routing and scheduling with working hour rules. *Transportation Science* **47**(1) 81–107. doi:10.1287/trsc.1120.0417.

Righini, G., M. Salani. 2006. Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization* **3**(3) 255 – 273. doi:10.1016/j.disopt.2006.05.007.

Solomon, M. 1987. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research* **35**(2) 254–265.

Tilk, C., A.-K. Rothenbächer, T. Gschwind, S. Irnich. 2017. Asymmetry matters: Dynamic half-way points in bidirectional labeling for solving shortest path problems with resource constraints faster. *European Journal of Operational Research* **261**(2) 530 – 539. doi:10.1016/j.ejor.2017.03.017.

Vidal, T., T.G. Crainic, M. Gendreau, C. Prins. 2014. A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research* **234**(3) 658 – 673. doi:10.1016/j.ejor.2013.09.045.

Xu, H., Z.-L. Chen, S. Rajagopal, S. Arunapuram. 2003. Solving a practical pickup and delivery problem. *Transportation Science* **37**(3) 347–364.

Zäpfel, G., M. Bögl. 2008. Multi-period vehicle routing and crew scheduling with outsourcing options. *International Journal of Production Economics* **113** 980–996.

Appendix: Backward Labeling

This appendix analyzes characteristics of the regulations which facilitate an efficient backward labeling method in which unnecessary label extensions are avoided.

United States

For REFs associated to break and rest periods we have

$$l_{\triangleleft} \preceq g^{\text{idle}}_{\Delta}(l_{\triangleleft}) \tag{13a}$$

$$g^{\text{rest}}_{t^{\text{rest}}}(l_{\triangleleft}) \preceq g^{\text{rest}}_{\Delta}(l_{\triangleleft}) \tag{13b}$$

$$g^{\text{break}}_{t^{\text{break}}}(l_{\triangleleft}) \preceq g^{\text{break}}_{\Delta}(l_{\triangleleft}). \tag{13c}$$

Therefore, REF g^{idle} will only generate dominated labels and can be ignored. Furthermore, breaks and rest periods shall always be scheduled with minimal duration.

Because of

$$g^{\text{rest}}_{t^{\text{rest}}}(l_{\triangleleft}) \preceq g^{\text{rest}}_{t^{\text{rest}}} \circ g^{\text{rest}}_{t^{\text{rest}}}(l_{\triangleleft}) \tag{14a}$$

$$g^{\text{rest}}_{t^{\text{rest}}}(l_{\triangleleft}) \preceq g^{\text{break}}_{t^{\text{break}}} \circ g^{\text{rest}}_{t^{\text{rest}}}(l_{\triangleleft}) \tag{14b}$$

$$g^{\text{rest}}_{t^{\text{rest}}}(l_{\triangleleft}) \preceq g^{\text{rest}}_{t^{\text{rest}}} \circ g^{\text{break}}_{t^{\text{break}}}(l_{\triangleleft}) \tag{14c}$$

$$g^{\text{break}}_{t^{\text{break}}}(l_{\triangleleft}) \preceq g^{\text{break}}_{t^{\text{break}}} \circ g^{\text{break}}_{t^{\text{break}}}(l_{\triangleleft}) \tag{14d}$$

there is no benefit in scheduling breaks and rests after another.

For any value $\Delta > 0$ we have

$$g^{\text{drive}}_{\max\{0,\Delta-\Delta^{\texttt{US}}_{l_{\triangleleft}}\}} \circ g^{\text{rest}}_{t^{\text{rest}}} \circ g^{\text{drive}}_{\min\{\Delta,\Delta^{\texttt{US}}_{l_{\triangleleft}}\}}(l_{\triangleleft}) \preceq g^{\text{drive}}_{\Delta} \circ g^{\text{rest}}_{t^{\text{rest}}}(l_{\triangleleft}) \tag{15a}$$

$$g^{\text{drive}}_{\max\{0,\Delta-\Delta^{\texttt{US}}_{l_{\triangleleft}}\}} \circ g^{\text{break}}_{t^{\text{break}}} \circ g^{\text{drive}}_{\min\{\Delta,\Delta^{\texttt{US}}_{l_{\triangleleft}}\}}(l_{\triangleleft}) \preceq g^{\text{drive}}_{\Delta} \circ g^{\text{break}}_{t^{\text{break}}}(l_{\triangleleft}). \tag{15b}$$

Hence, we can conclude that no break or rest activities are scheduled if $\Delta^{\texttt{US}}_{l_{\triangleleft}} > 0$ and that driving periods are always scheduled with duration $\Delta^{\texttt{US}}_{l_{\triangleleft}}$.

Lastly, if $l^{\text{trip}}_{\triangleleft} = 0$ and $l^{\text{time}}_{\triangleleft} \leq t^{\max}_n + s_n$ we have

$$g^{\text{rest}}_{\Delta} \circ g^{\text{visit}}_{n}(l_{\triangleleft}) \preceq g^{\text{visit}}_{n} \circ g^{\text{rest}}_{\Delta}(l_{\triangleleft}) \tag{16a}$$

$$g^{\text{break}}_{\Delta} \circ g^{\text{visit}}_{n}(l_{\triangleleft}) \preceq g^{\text{visit}}_{n} \circ g^{\text{break}}_{\Delta}(l_{\triangleleft}) \tag{16b}$$

$$\tag{16c}$$

Thus, if $g^{\text{visit}}_n(l_{\triangleleft})$ is feasible and $l^{\text{time}}_{\triangleleft} \leq t^{\max}_n + s_n$, there is no benefit in scheduling a break or rest before the service.

European Union

For REFs associated to off-duty periods we have

$$
\begin{aligned}
l_{\triangleleft} &\preceq g^{\text{idle}}_{\Delta}(l_{\triangleleft}) && \text{(17a)}\\
g^{\text{dayrest}}_{t^{\text{rest}}}(l_{\triangleleft}) &\preceq g^{\text{dayrest}}_{\Delta}(l_{\triangleleft}) && \text{(17b)}\\
g^{\text{dayrest|2nd}}_{t^{\text{rest|2nd}}}(l_{\triangleleft}) &\preceq g^{\text{dayrest|2nd}}_{\Delta}(l_{\triangleleft}) && \text{(17c)}\\
g^{\text{nightrest}}_{t^{\text{rest}}}(l_{\triangleleft}) &\preceq g^{\text{nightrest}}_{\Delta}(l_{\triangleleft}) && \text{(17d)}\\
g^{\text{nightrest|2nd}}_{t^{\text{rest|2nd}}}(l_{\triangleleft}) &\preceq g^{\text{nightrest|2nd}}_{\Delta}(l_{\triangleleft}) && \text{(17e)}\\
g^{\text{rest|1st}}_{t^{\text{rest|1st}}}(l_{\triangleleft}) &\preceq g^{\text{rest|1st}}_{\Delta}(l_{\triangleleft}) && \text{(17f)}\\
g^{\text{break|2nd}}_{t^{\text{break|2nd}}}(l_{\triangleleft}) &\preceq g^{\text{break|2nd}}_{\Delta}(l_{\triangleleft}) && \text{(17g)}\\
g^{\text{break|1st}}_{t^{\text{break|1st}}}(l_{\triangleleft}) &\preceq g^{\text{break|1st}}_{\Delta}(l_{\triangleleft}). && \text{(17h)}
\end{aligned}
$$

Thus, REF g^{idle} can be neglected and when applying any of the other REFs, we can use the minimum duration required by the regulation and larger values of Δ do not have to be considered. Using any of these REFs multiple times after another may only be relevant for $g^{\text{nightrest}}$, because $g^{a}_{\Delta_2} \circ g^{a}_{\Delta_1}(l_{\triangleleft})$ is infeasible for $a \in \{\text{dayrest}, \text{dayrest|2nd}, \text{nightrest|2nd}, \text{rest|1st}, \text{break|2nd}, \text{break|1st}\}$. As

$$
\begin{aligned}
g^{\text{dayrest}}_{\Delta_1+\Delta_2}(l_{\triangleleft}) &\preceq g^{\text{rest|1st}}_{\Delta_2} \circ g^{\text{dayrest|2nd}}_{\Delta_1}(l_{\triangleleft}) && \text{(18a)}\\
g^{\text{nightrest}}_{\Delta_1+\Delta_2}(l_{\triangleleft}) &\preceq g^{\text{rest|1st}}_{\Delta_2} \circ g^{\text{nightrest|2nd}}_{\Delta_1}(l_{\triangleleft}) && \text{(18b)}
\end{aligned}
$$

for any value $\Delta_1, \Delta_2 > 0$ we can conclude that the first part of a rest is never taken immediately before the second part. Furthermore, rest periods taken during a day are never needed before or after a night rest because of

$$
\begin{aligned}
g^{\text{nightrest}}_{\Delta_2}(l_{\triangleleft}) &\preceq g^{\text{nightrest}}_{\Delta_2} \circ g^{\text{dayrest}}_{\Delta_1}(l_{\triangleleft}) && \text{(19a)}\\
g^{\text{nightrest|2nd}}_{\Delta_2}(l_{\triangleleft}) &\preceq g^{\text{nightrest|2nd}}_{\Delta_2} \circ g^{\text{dayrest}}_{\Delta_1}(l_{\triangleleft}) && \text{(19b)}\\
& && \text{(19c)}
\end{aligned}
$$

and

$$
\begin{aligned}
g^{\text{nightrest}}_{\Delta_1}(l_{\triangleleft}) &\preceq g^{\text{dayrest}}_{\Delta_2} \circ g^{\text{nightrest}}_{\Delta_1}(l_{\triangleleft}) && \text{(20a)}\\
g^{\text{nightrest}}_{\Delta_1}(l_{\triangleleft}) &\preceq g^{\text{dayrest|2nd}}_{\Delta_2} \circ g^{\text{nightrest}}_{\Delta_1}(l_{\triangleleft}). && \text{(20b)}
\end{aligned}
$$

For any label $l_{\triangleleft}$ with $l^{\text{work|B}}_{\triangleleft} = 0$ we have

$$
h(l_{\triangleleft}) \preceq g^{\text{break|1st}}_{\Delta} \circ h \circ g^{\text{break|2nd}}_{\Delta}(l_{\triangleleft}) \qquad \text{(21)}
$$

where h represents any sequence of the REFs g^{drive} and g^{visit}. Therefore, we can assume that the second part of a break is only taken if $l_{\triangleleft}^{\text{work|B}} > 0$.

For any value $\Delta > 0$ we have

$$g^{\text{drive}}_{\max\{0,\Delta-\Delta^{\mathtt{EU}}_{l_{\triangleleft}}\}} \circ h \circ g^{\text{drive}}_{\min\{\Delta,\Delta^{\mathtt{EU}}_{l_{\triangleleft}}\}}(l_{\triangleleft}) \preceq g^{\text{drive}}_{\Delta} \circ h(l_{\triangleleft}) \tag{22}$$

where h represents any sequence of REFs not including g^{drive} and g^{visit}. Hence, we can conclude that no break or rest activities are scheduled if $\Delta^{\mathtt{EU}}_{l_{\triangleleft}} > 0$ and that driving periods are always scheduled with duration $\Delta^{\mathtt{EU}}_{l_{\triangleleft}}$.

Lastly, if $l_{\triangleleft}^{\text{trip}} = 0$ and $l_{\triangleleft}^{\text{time}} \leq t_n^{\max} + s_n$ we have

$$g^{a}_{\Delta} \circ g^{\text{visit}}_{n}(l_{\triangleleft}) \preceq g^{\text{visit}}_{n} \circ g^{a}_{\Delta}(l_{\triangleleft}) \tag{23}$$

for $a \in \{\text{nightrest}, \text{nightrest|2nd}, \text{dayrest}, \text{dayrest|2nd}, \text{rest|1st}, \text{break|2nd}, \text{break|1st}\}$. Thus, if $g^{\text{visit}}_{n}(l_{\triangleleft})$ is feasible and $l_{\triangleleft}^{\text{time}} \leq t_n^{\max} + s_n$, there is no benefit in scheduling any other activity before the service.

Appendix: Detailed Computational Results

Tables 15-17 show the results for the 25 and 50 customer instances with an average speed of 70 km/h and a cost structure of 0.50 Euro per kilometer. Regarding the EU regulations, a daily cost of $c^{\text{day}} = 150$ Euro is used. For each instance the computation time (in seconds) and the costs of the best solution found are shown.

	25 customer		50 customer	
instance	Time[s]	Costs	Time[s]	Costs
TDS_C101	0.10	191.167	0.811	362.167
TDS_C102	9.18	190.083	10.248	361.083
TDS_C103	60.51	189.417	613.715	360.417
TDS_C104	4677.99	186.667	7200	358.167
TDS_C105	0.37	191.167	2.231	362.167
TDS_C106	0.19	191.167	1.684	362.167
TDS_C107	0.89	191.167	6.006	362.167
TDS_C108	3.23	189.75	23.852	360.75
TDS_C109	52.74	187.833	443.53	358.833
TDS_C201	4.52	248	767.943	434.75
TDS_C202	3.00	217.833	7200	392
TDS_C203	17.57	217.833	2313.08	362.75
TDS_C204	137.28	214.167	7200	365.417
TDS_C205	1.14	214.417	7200	369.417
TDS_C206	1.78	214.417	7200	369.417
TDS_C207	41.01	214.167	7200	374.333
TDS_C208	6.74	214.167	7200	369.417
TDS_R101	0.27	502.25	2.137	847.833
TDS_R102	0.44	446.25	41.698	753.917
TDS_R103	1.95	401.083	2322.08	649.083
TDS_R104	1.54	359.417	7200	536.417
TDS_R105	0.56	438.167	160.177	749.583
TDS_R106	1.28	407.083	2970.37	687.667
TDS_R107	10.25	391.833	6136.73	610.583
TDS_R108	233.57	349.417	7200	540.833
TDS_R109	6.43	385.083	7200	645.167
TDS_R110	76.05	354.417	3122.41	571.917
TDS_R111	8.33	387.667	7200	605.75
TDS_R112	1796.10	337.333	7200	529.417
TDS_R201	0.45	463.583	16.458	798.917
TDS_R202	0.75	410.75	373.988	714.333
TDS_R203	5.71	391.833	4875.74	626
TDS_R204	97.31	355.167	7200	528.5
TDS_R205	2.43	404.083	176.199	695.25
TDS_R206	15.61	378.083	7200	643.75
TDS_R207	3.76	367.167	7200	594.167
TDS_R208	90.62	341.083	7200	534.5
TDS_R209	8.99	376.75	2083.59	615.583
TDS_R210	14.13	411.75	7200	676.417
TDS_R211	40.81	351.167	7200	581.083
TDS_RC101	0.90	358.25	4.196	632.583
TDS_RC102	2.37	335.917	48.078	604.417
TDS_RC103	5.13	327.083	899.042	584.667
TDS_RC104	129.71	299.75	2542.88	522.917
TDS_RC105	0.64	334.75	26.301	613.75
TDS_RC106	6.80	310.833	63.194	564.917
TDS_RC107	124.94	296.333	839.544	522.667
TDS_RC108	825.24	294.5	6618.75	517.667
TDS_RC201	0.22	360.5	5.242	684.833
TDS_RC202	1.33	338.167	14.414	613.833
TDS_RC203	4.83	327.083	65.644	594.917
TDS_RC204	35.45	299.75	7200	491.917
TDS_RC205	0.37	338.083	10.405	631.833
TDS_RC206	0.69	324.25	25.911	610.167
TDS_RC207	5.59	298.333	209.52	560
TDS_RC208	1650.98	294.5	7200	517.667

Table 15 Detailed results for the US regulations

Instance	Time[s]			Costs		
	20h–7h	23h–6h	0h–4h	20h–7h	23h–6h	0h–4h
TDS_RC101	0.406	1.754	1.245	4,618.83	4,461.25	4,347.42
TDS_RC102	2.003	30.32	132.836	4,324.75	4,161.92	4,024.75
TDS_RC103	4.815	61.414	441.645	4,100.08	3,950.08	3,943.67
TDS_RC104	34.292	824.184	414.33	3,949.00	3,668.83	3,640.83
TDS_RC105	2.73	5.303	25.583	4,378.42	4,218.50	4,218.50
TDS_RC106	1.934	24.398	22.417	3,877.17	3,877.17	3,842.75
TDS_RC107	50.424	733.44	3126.69	3,424.33	3,424.33	3,424.33
TDS_RC108	219.125	2556.38	3680.45	3,411.50	3,411.50	3,411.50
TDS_RC201	0.281	0.608	1.107	5,113.75	5,103.83	5,089.25
TDS_RC202	0.873	9.126	17.721	4,631.17	4,534.25	4,477.67
TDS_RC203	4.992	16.333	21.871	4,255.33	4,255.33	4,255.33
TDS_RC204	196.092	894.305	952.914	3,959.92	3,940.67	3,940.67
TDS_RC205	1.357	4.306	3.977	4,631.17	4,616.58	4,441.42
TDS_RC206	0.842	6.411	5.897	4,428.00	4,390.08	4,276.92
TDS_RC207	3.01	8.549	76.814	3,738.33	3,738.33	3,738.33
TDS_RC208	82.706	1353.91	3802.44	3,308.17	3,254.50	3,254.50
TDS_C101	0.702	2.464	16.551	3,400.08	3,400.08	3,313.75
TDS_C102	7200	7200	1909.2	2,979.33	2,951.33	2,824.67
TDS_C103	7200	7200	3847.15	2,846.25	2,800.67	2,641.42
TDS_C104	7200	7200	7200	2,644.92	2,615.08	2,481.50
TDS_C105	0.67	1.388	6.911	3,313.75	3,163.75	3,163.75
TDS_C106	0.874	3.198	1.544	3,400.08	3,400.08	3,313.75
TDS_C107	36.238	8.081	9.563	3,163.75	2,961.83	2,961.83
TDS_C108	60.465	202.844	384.361	2,961.83	2,961.83	2,930.25
TDS_C109	279.8	6287.09	7200	2,633.25	2,581.25	2,581.25
TDS_C201	0.936	13.166	23.15	4,618.92	3,680.08	3,261.08
TDS_C202	27.003	160.896	7200	3,702.25	3,447.83	2,924.42
TDS_C203	133.254	7200	7200	3,380.08	3,065.00	3,062.67
TDS_C204	1406.63	7200	7200	2,669.92	2,771.42	2,779.58
TDS_C205	555.653	2.808	3.666	3,321.17	2,850.92	2,850.92
TDS_C206	10.124	7.8	18.533	2,850.92	2,850.92	2,850.92
TDS_C207	88.95	7200	453.798	2,849.17	2,850.92	2,700.92
TDS_C208	5.476	27.768	32.292	2,850.92	2,849.17	2,849.17
TDS_R101	0.109	1.232	7.394	7,629.08	7,078.17	6,742.67
TDS_R102	2.355	28.329	40.668	6,672.58	5,831.08	5,831.08
TDS_R103	9.219	28.828	103.739	5,154.92	4,869.58	4,869.58
TDS_R104	12.916	1636.11	2365.98	4,436.08	4,347.83	4,342.58
TDS_R105	1.654	1.544	7.659	5,656.67	5,312.33	5,300.75
TDS_R106	1.591	21.075	144.25	4,965.83	4,876.58	4,834.50
TDS_R107	37.346	291.669	168.134	4,611.67	4,571.92	4,421.92
TDS_R108	156.513	421.272	646.206	4,182.17	3,910.83	3,902.08
TDS_R109	45.521	84.13	62.664	4,811.75	4,669.33	4,532.25
TDS_R110	174.485	256.591	1517.87	4,159.42	3,928.33	3,928.33
TDS_R111	27.003	48.411	110.181	4,605.17	4,523.50	4,373.50
TDS_R112	500.507	4345.74	7200	3,808.17	3,767.33	3,808.17
TDS_R201	2.106	6.256	12.48	5,963.00	5,963.00	5,887.17
TDS_R202	10.873	14.633	15.038	5,265.08	4,860.17	4,860.17
TDS_R203	23.088	40.872	52.196	4,683.92	4,604.58	4,604.58
TDS_R204	293.949	259.783	349.404	4,041.00	3,975.58	3,975.58
TDS_R205	7.332	42.431	33.29	5,245.33	5,017.17	4,883.00
TDS_R206	158.511	888.346	523.404	4,825.75	4,762.67	4,598.75
TDS_R207	34.834	449.602	93.474	4,407.92	4,407.92	4,323.92
TDS_R208	231.705	2788.4	7200	3,895.08	3,895.08	3,795.92
TDS_R209	14.133	220.956	700.169	4,456.42	4,414.92	4,380.50
TDS_R210	31.948	97.265	26.052	4,732.33	4,706.67	4,636.08
TDS_R211	11.185	141.708	311.045	3,808.17	3,808.17	3,808.17

Table 16 Detailed results for the EU regulations and 25 customer instances

Instance	Time[s]			Costs		
	20h–7h	23h–6h	0h–4h	20h–7h	23h–6h	0h–4h
TDS_RC101	3.291	21.2	84.645	8,561.92	7,942.83	7,927.08
TDS_RC102	14.383	555.946	329.842	7,926.00	7,579.92	7,568.25
TDS_RC103	141.366	2339.03	4031.44	7,509.33	7,058.17	7,023.75
TDS_RC104	2616.04	7200	7200	6,568.75	6,449.67	6,449.67
TDS_RC105	12.276	251.016	350.38	8,466.25	7,820.33	7,641.17
TDS_RC106	71.617	831.859	875.173	7,062.25	7,062.25	6,946.75
TDS_RC107	734.531	3394.86	7196.3	6,261.17	6,215.67	6,215.67
TDS_RC108	2049.81	7200	7200	6,023.67	6,023.67	6,023.67
TDS_RC201	10.795	22.588	20.926	10,475.30	9,750.58	9,584.83
TDS_RC202	41.262	202.937	263.071	9,384.25	9,195.75	9,045.75
TDS_RC203	412.852	819.239	1238.73	8,371.17	8,352.50	8,347.25
TDS_RC204	7200	7200	7200	5,849.75	5,806.00	5,910.42
TDS_RC205	47.876	652.072	151.708	9,233.58	9,112.83	8,375.33
TDS_RC206	49.249	337.143	260.689	8,098.75	7,932.42	7,780.08
TDS_RC207	133.239	713.223	1348.07	7,011.00	6,865.67	6,844.08
TDS_RC208	2960.07	7200	7200	6,032.92	5,947.75	5,968.17
TDS_C101	0.983	7.348	11.014	6,285.17	6,285.17	6,135.17
TDS_C102	136.952	2950.34	7200	5,851.50	5,691.42	5,691.42
TDS_C103	1223.69	7200	7200	5,678.58	5,576.42	5,555.92
TDS_C104	7200	7200	7200	4,740.75	4,741.92	4,767.58
TDS_C105	2.824	30.186	169.648	6,135.17	5,928.50	5,928.50
TDS_C106	4.696	22.697	30.217	5,985.17	5,985.17	5,985.17
TDS_C107	3039.81	7200	7200	5,928.50	5,766.75	5,819.75
TDS_C108	7200	7200	7200	5,669.83	5,613.00	5,482.58
TDS_C109	7200	7200	7200	5,029.08	4,825.42	4,761.83
TDS_C201	11.653	349.732	7200	7,344.42	6,179.17	5,935.75
TDS_C202	92.242	2619.66	7200	6,008.75	5,729.25	5,450.25
TDS_C203	330.063	7200	7200	5,851.75	5,652.75	5,044.17
TDS_C204	7200	7200	7200	4,968.75	4,604.67	4,754.08
TDS_C205	7200	7200	7200	5,529.67	5,505.75	5,544.25
TDS_C206	7200	7200	7200	5,457.33	5,187.08	5,168.42
TDS_C207	7200	7200	7200	4,786.25	4,796.75	4,794.42
TDS_C208	7200	7200	7200	5,071.00	4,794.42	4,794.42
TDS_R101	1.544	16.363	30.217	12,875.10	12,398.40	11,584.80
TDS_R102	7.941	119.968	78.795	11,372.20	10,712.30	10,065.10
TDS_R103	1146.84	822.534	1057.62	8,912.00	8,398.50	8,368.75
TDS_R104	7200	7200	7200	6,729.58	6,814.25	6,737.08
TDS_R105	29.92	606.083	1477.47	9,038.33	8,814.83	8,719.67
TDS_R106	582.672	7200	6802.17	8,223.17	8,042.33	7,863.17
TDS_R107	6455.99	7200	7200	7,348.67	7,294.33	7,006.08
TDS_R108	7200	7200	7200	6,585.50	6,532.92	6,641.50
TDS_R109	1758.9	7200	7200	7,561.50	7,416.25	7,422.67
TDS_R110	7200	7200	7200	6,895.83	6,859.00	6,823.50
TDS_R111	7200	7200	7200	7,283.83	7,299.08	7,003.75
TDS_R112	7200	7200	7200	6,211.42	6,331.67	6,373.67
TDS_R201	26.473	163.58	250.065	10,643.60	10,266.00	9,836.42
TDS_R202	1167.78	4047.53	7200	9,372.83	9,057.75	8,732.00
TDS_R203	1122.24	7200	7200	8,178.42	8,149.83	7,650.25
TDS_R204	7200	7200	7200	6,158.92	6,501.42	6,349.08
TDS_R205	1197.21	7200	7200	8,585.50	8,285.00	8,157.83
TDS_R206	2266.9	7200	7200	7,527.08	7,631.50	7,575.00
TDS_R207	7200	7200	7200	6,881.25	6,761.00	6,940.17
TDS_R208	7200	7200	7200	5,900.42	6,214.33	5,943.00
TDS_R209	5205.48	7200	7200	7,400.50	7,153.08	7,184.58
TDS_R210	7200	7200	7200	7,857.33	7,702.75	7,559.17
TDS_R211	7200	7200	7200	6,656.00	6,697.42	6,671.17

Table 17 **Detailed results for the EU regulations and 50 customer instances**