An Exact Method for Vehicle Routing and Truck Driver Scheduling Problems

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

Stefan Irnich
Chair of Logistics Management, Gutenberg School of Management and Economics,
Johannes Gutenberg University Mainz, Mainz, Germany
irnich@uni-mainz.de

In most developed countries working hours of truck drivers are constrained by hours of service regulations. When optimizing vehicle routes, trucking companies must consider these constraints in order to assure that drivers can comply with the regulations. This paper studies the combined vehicle routing and truck driver scheduling problem (VRTDSP), which generalizes the well-known vehicle-routing problem with time windows by considering working hour constraints. A branch-and-price algorithm for solving the VRTDSP is presented. This is the first algorithm that solves the VRTDSP to proven optimality.

Key words: Hours of Service Regulations; Vehicle Routing; Truck Driver Scheduling; Branch-and-Price

1. Introduction

In long-distance haulage, truck drivers usually spend a large amount of their time driving. Without regularly taking breaks and rest periods drivers would be exposed to unnecessarily high risks of fatigue-related accidents. To increase road safety, many governments world wide impose hours of service regulations for truck drivers limiting the amount of driving and working. Governments herewith stipulate that a minimum amount of break and rest time is taken throughout a trip. In the United States, for example, new hours of service regulations entered into force in 2013 (Federal Motor Carrier Safety Administration 2011). According to these regulations, a driver must not drive for more than eleven hours without taking a rest period of at least ten 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 eight hours have elapsed since the end of the last rest or break period of at least 30 minutes. The most recent condition that prohibits driving without taking a break of at least 30 minutes was introduced with the 2013 rule change. Prior to the rule change, a driver was allowed to drive up to eleven hours without a break (Federal Motor Carrier Safety Administration 2008).

Transport companies seek to reduce costs by optimizing vehicle routes. On an operational level, this requires the minimization of the total distance traveled by all vehicles, while various operational constraints such as capacity and time window constraints must be considered. The resulting decision problem is known as the *vehicle routing problem with time windows* (VRPTW) or a variant thereof (Cordeau et al. 2002, Desaulniers et al. 2014). Interestingly, the vast majority of the VRPTW-related literature does not consider hours of service regulations. As a result, routes computed by any of the proposed solution approaches are likely to be infeasible in a long-distance haulage context, where hours of service regulations must be complied with. In the last years, the transportation science community has increasingly attempted to close this gap between academic research and practical requirements. Specifically for variants of the VRPTW with service regulations, several heuristic approaches have been developed. So far, however, no exact approach has been presented. Due to this void, the quality of heuristics can only be assessed in relation to one and the other, but not in an absolute way. This paper provides the first exact algorithm able to solve the *vehicle routing and truck driver scheduling problem* (VRTDSP). More precisely, we present a branch-and-price algorithm for a variant of the VRPTW extended by the constraints imposed by applicable hours of service regulations.

The main contribution of the paper at hand is the development of an exact algorithm for the VRTDSP that is based on an auxiliary network allowing to explicitly model all possible driver activities. In this auxiliary network, driver activities correspond to arcs and the time to traverse such an arc depends on a parameter indicating the duration of the activity. This paper shows that, for hours of service regulations in the United States, it is possible to replace this auxiliary network by a parameter-free network in which the duration of each activity can be uniquely determined. Furthermore, a similar parameter-free network can be derived for European Union regulations. The parameter-free networks allow us to efficiently solve the resulting (elementary) shortest-path problem with resource constraints using labeling-based solution approaches. We develop such an efficient labeling algorithm and prove its practical applicability in a computational study, in which optimal solutions for all VRTDSP instances with 25 customers are obtained for U.S. hours of service regulations and 53 of 56 instances are solved for EU regulations.

The remainder of this paper is structured as follows. Section 2 presents a brief overview of hours of service constraints in the vehicle routing literature. The VRTDSP is formally introduced in Section 3. Section 4 presents the general solution framework and details how U.S. hours of service regulations can be modeled in an extended network using resource extension functions. We develop dominance rules allowing to model the subproblem in such a way that labeling approaches can solve the problem efficiently. Moreover, we show how the performance of the proposed branch-and-price algorithm can be further improved and other regulations, in particular EU regulations, can be considered. Computational experiments and their results are presented in Section 5 before concluding remarks are given in Section 6.

2. Hours of Service Regulations in the Vehicle Routing Literature

An early work explicitly considering hours of service regulations is presented by Xu et al. (2003), who study a rich vehicle routing problem considering multiple time windows and U.S. hours of service regulations. The authors conjecture that the problem of minimizing total costs of all on- and off-duty times for a given tour is $\mathcal{NP}$-hard in the presence of U.S. hours of service regulations. Moreover, they present a column generation approach based on a heuristic for scheduling on- and off-duty periods.

For previous hours of service regulations in the United States, Archetti and Savelsbergh (2009) show that, in the case of single time windows, the problem of determining a feasible truck driver schedule for a given tour can be solved in $\mathcal{O}(k^3)$, where k denotes the number of locations in the tour. Goel and Kok (2012b) show that this problem can be solved in $\mathcal{O}(k^2)$ and that the complexity does not increase for multiple time windows if the time between two successive time windows is at least 10 hours, i.e., the minimum length of a rest period. This can be the case if time windows are tied to business hours, e.g., if customers request to be visited on any day between 8 AM and 8 PM. A generic model capable of representing various regulations, including U.S. hours of service regulations, is presented in Goel (2012). In Goel (2014) a simple heuristic approach for the VRTDSP in the United States is used to analyze the impact of the recent rule change on costs and accident risks.

For hours of service regulations in Europe, Canada, and Australia, the problem of determining feasible truck driver schedules has been studied by Goel (2010), Drexl and Prescott-Gagnon (2010), Goel and Rousseau (2012), and Goel et al. (2012), respectively. Due to the various complicating constraints in these regulations, the complexity of finding a feasible schedule is also unknown. Only for the special case of team driving, it is known that a feasible schedule complying with EU regulations can be determined in quadratic time (Goel and Kok 2012a).

Column generation techniques have been used to heuristically solve vehicle routing problems with constraints on driver schedules. One of the early works explicitly considering breaks and night rests within a vehicle routing context is presented by Savelsbergh and Sol (1998). In this work, drivers must have a 45 minute lunch break every day between 11 AM and 2 PM and a night rest between any two working days. Although not explicitly mentioned, the break and rest requirements most likely have the purpose of generating truck driver schedules complying with EU regulations that were in force at the time. The requirements that breaks and rest periods must be taken within fix time intervals is stricter than demanded by EU regulations. These stricter rules have the advantage that the size of the search space of the resulting decision problem is reduced. A branch-and-price approach is used to heuristically solve this problem in a dynamic environment, in which new transportation requests can arrive at any time.

With the introduction of new hours of service regulations in the European Union in 2007, several heuristic approaches for combined vehicle routing and truck driver scheduling have been proposed, e.g., by Zäpfel and Bögl (2008), Goel (2009), Ceselli et al. (2009), Bartodziej et al. (2009), Prescott-Gagnon et al. (2010), Kok et al. (2010), and Derigs et al. (2011). U.S. hours of service regulations have so far found little attention in the vehicle routing literature. Recently, Rancourt et al. (2013) presented a tabu search approach for the U.S. hours of service regulations in force until July 2013. Goel and Vidal (2014) present a hybrid genetic search for vehicle routing and truck driver scheduling, which has been evaluated for various different regulations world wide. Amongst others this approach can generate routes and schedules complying with the new regulations in the United States.

3. The Vehicle Routing and Truck Driver Scheduling Problem

The VRTDSP can briefly be defined as a VRPTW, in which routes are scheduled so that each driver can comply with hours of service regulations. More formally, let C be the set of customers and let n^{depot} denote the depot, i.e., the start and end of a route. Furthermore, let $N = C \cup \{n^{\text{depot}}\}$ denote the set of all nodes and let $A = \{(n,m) \in N \times N : n \neq m\}$ denote the set of arcs between these nodes in the network $\mathcal{G} = (N, A)$. For each node $n \in C$, a time window $[t_n^{\min}, t_n^{\max}]$, a demand q_n, and a service time s_n are given. For simplicity, the same notation is used for the depot, where demand and service time are assumed to be zero and the time window spans the full planning horizon.

For each arc $(n,m) \in A$, travel costs c_{nm} and the driving time d_{nm} (without break or rest period) is given. Obviously, all arcs that trivially cannot be used in a feasible solution, e.g., due to incompatible time windows, can be removed from the set of arcs. A sufficiently large fleet of homogeneous vehicles is stationed at the depot, each having identical capacity Q. For the sake of convenience, all coefficients are assumed to be non-negative integers.

A *route* is a walk $r = (n_0, n_1, \ldots, n_{k-1}, n_k)$ in the network $\mathcal{G} = (N, A)$, where $n_0 = n_k = n^{\text{depot}}$, and $n_i \in C$ for all $0 < i < k$. A route is *feasible* if $\sum_{i=1}^{k-1} q_{n_i} \leq Q$ and if a schedule complying with hours of service regulation exists in which the service at each customer $n \in C$ begins within the time window $[t_n^{\min}, t_n^{\max}]$.

The *cost* of route r is $c_r = \sum_{i=1}^{k} c_{n_{i-1},n_i}$. The VRTDSP is the problem of finding a set of feasible routes such that each customer in C is visited exactly by one route and that total costs are minimized.

Whether a schedule complying with hours of service regulations exists depends on the specific rules imposed by the regulation. Obviously, it is impossible to give a comprehensive overview of all national hours of service regulations world wide within this paper. Thus, the next sections focus on hours of service regulations in the United States, whereas Section 4.5 describes how the methodologies can be adapted to different regulations, in particular European Union regulations. Table 1 summarizes the parameters of hours of service regulations in the United States. According to these regulations, rest and break periods can be scheduled at any time and with any duration of at least t^{rest} or t^{break}, respectively. Whether driving periods may be scheduled, however, depends on the drivers' state. We assume service times at customer locations to be work periods which must not be interrupted by breaks or rests. A detailed model and approach to determine a driver schedule complying with the regulations and satisfying time window constraints is described in Sections 4.1 and 4.2.

Symbol	**Value**	**Description**
$t^{\text{drive}\|\text{R}}$	11 hours	The maximum accumulated driving time between two consecutive rest periods
t^{break}	$\frac{1}{2}$ hours	The minimum duration of a break period
t^{rest}	10 hours	The minimum duration of a rest period
$t^{\text{elapsed}\|\text{B}}$	8 hours	The maximum time after the end of the last break or rest period until which a driver may drive
$t^{\text{elapsed}\|\text{R}}$	14 hours	The maximum time after the end of the last rest period until which a driver may drive

Table 1 **Parameters imposed by the new U.S. hours of service regulations**

The requirement concerning break periods has been introduced with the recent rule change in 2013 and previous regulations can be interpreted as a relaxed version of the current regulations resulting from setting $t^{\text{elapsed}|\text{B}} = \infty$, so that there is no need for breaks.

Assuming that we are given the set of all feasible routes R, the VRTDSP can be modeled using the following set partitioning formulation:

$$\min \sum_{r \in R} c_r \lambda_r \tag{1a}$$

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

$$\lambda_r \in \{0, 1\} \qquad \text{for all } r \in R \tag{1c}$$

In this model, the binary route variables λ_r indicate, which routes $r \in R$ are selected in the solution. The objective (1a) is to minimize the cumulative costs of all routes. The coefficients a_{nr} state how often a route r visits a customer $n \in C$. Hence, constraints (1b) ensures that each customer is visited by exactly one route. The domain of the binary variables is given by (1c).

Note that non-elementary routes, i.e., routes that visit one or several customers more than once, are never part of an integer solution due to the definition of the coefficients a_{nr} and the partitioning requirement. Moreover, as we assume that the triangle inequality for travel times and costs holds, an optimal solution to the VRTDSP visits each customer exactly once, even if multiple visits were allowed, so that covering constraints can replace the partitioning constraints (1b).

4. Branch-and-Price

The currently leading methods for solving many variants of the *vehicle routing problem* (VRP) to proven optimality are based on column generation techniques (Lübbecke and Desrosiers 2005, Desaulniers et al. 2005). Typically, the starting point is a compact formulation of the VRP, to which a Dantzig-Wolfe decomposition is applied. The result is the so-called extensive formulation or master program, which is often a set partitioning type of model such as problem (1). Indeed, this set partitioning formulation is a versatile model for all those VRP variants, in which the only coupling constraints are related to serving each customer once by one vehicle. The set R implicitly models that each route $r \in R$ must satisfy all operational constraints.

Branch-and-price is the solution of an extensive model via column generation techniques inside a branch-and-bound method. It works for both the VRPTW and the VRTDSP as follows: First, the linear relaxation of (1) is solved with a column generation algorithm. Starting with a restricted master program (RMP) containing only a subset of the route variables $\lambda_r \geq 0$, the RMP is optimized. Its dual solution determines the column generation subproblem, the so-called pricing problem, which asks for the determination of a route r with negative reduced costs. The associated variable λ_r is then added to the RMP, and the process alternates between RMP re-optimization and pricing as long as negative reduced cost routes exist. Second, branching is required if the solution of the RMP is fractional. Branch-and-price, even if not named so, was proposed by Desrochers et al. (1992) and has become a well established method for solving the VRPTW. Excellent tutorials and surveys are (Desaulniers et al. 1998, Cordeau et al. 2002, Feillet 2010, Desaulniers et al. 2010, Baldacci et al. 2012).

The VRPTW and VRTDSP differ by the type of pricing problem that generates new feasible routes. In both cases, the pricing problem is a variant of the *shortest path problem with resource constraints* (SPPRC, Irnich and Desaulniers 2005). While only three resources (cost, time, and load) need to be considered for the VRPTW, the type of resources and their propagation along the path is not obvious for the VRTDSP. We will use seven and nine resources (see Sections 4.1 and 4.2) and build an extended auxiliary network to propagate these resources. Moreover, resources in the VRTDSP are intertwined, meaning that they partially depend on another in a non-linear way. To be able to efficiently solve the SPPRC we will, therefore, use tailored resource extension functions (Desaulniers et al. 1998, Irnich 2008).

Variants of the SPPRC are typically solved with dynamic programming labeling algorithms. Two major research directions are relevant here. First, modeling the VRTDSP with the help of resources requires a careful choice of resource extension functions (REFs). Important is the concept of non-decreasing REFs because for these it is simpler to derive and prove dominance rules (Desaulniers et al. 1998, Irnich and Desaulniers 2005). Note that strong dominance is crucial for a fast solution of SPPRCs. Second, a solution of the VRTDSP consists of elementary routes, but the partitioning model allows also non-elementary route variables. While the distinction makes no difference for the final integer solution, the different linear relaxations of the master program can produce very different bounds. For the non-elementary SPPRC, there exist pseudo-polynomial labeling algorithms (Desrochers 1986), but linear relaxation bounds are often weak. Generally, much stronger bounds result from the solution of the elementary version of the SPPRC, the so-called ESPPRC, but this problem is $\mathcal{NP}$-hard in the strong sense (Dror 1994). Over the years, intensive research has been spent on finding relaxations between ESPPRC and SPPRC, which provide a good tradeoff between (practical and theoretical) hardness of the problem and tightness of the bounds produced by the respective linear relaxation of the master program: Irnich and Villeneuve (2006) proposed a pseudo-polynomial labeling algorithm for SPPRC without k-cycles for fixed $k \geq 3$, Desaulniers et al. (2008) introduced partial elementary routes, and Baldacci et al. (2011a,b) invented the *ng*-route relaxation. Labeling algorithms for ESPPRC are often based on ideas presented in (Feillet et al. 2004, Boland et al. 2006, Righini and Salani 2008).

In Sections 4.1 and 4.2, we present REFs that can be used to determine whether a feasible schedule complying with U.S. hours of service regulations exists for a given route. Section 4.3 proposes several techniques for finding routes with negative reduced costs and Section 4.4 gives a brief outlook on possible extensions. The required adaptations for EU regulations will be discussed in Section 4.5.

4.1. Modeling Driver Activities with REFs

In the VRPTW it is not necessary to explicitly model driver activities. To ensure time-feasibility of a route it is sufficient to validate that the time between departure at customer n and the start of service at customer m must be at least d_{nm}. If more time is available, the driver can wait and the specific timing of driving and waiting periods can be arbitrarily set. In the VRTDSP, however, driver activities must be modeled explicitly. For any feasible route through the network $\mathcal{G} = (N, A)$, a schedule specifying the exact timing of all driver activities must be found. This schedule must comply with hours of service regulations and time window constraints. Each driver activity changes the state of the driver, which can be represented by a resource tuple that is modified with different REFs.

The fundamental idea allowing to explicitly model driver activities is that we can use an auxiliary network $\mathcal{G}' = (N', A')$ that is obtained by expanding the original network $\mathcal{G} = (N, A)$. The auxiliary network is created in such a way that each change of the resource tuple is the result of applying a REF associated to an arc in the auxiliary network $\mathcal{G}' = (N', A')$. More precisely, each arc (n, m) of the original arc set A is replaced by an auxiliary subnetwork $\mathcal{G}'_{nm}$. The subnetwork $\mathcal{G}'_{nm}$ has a unique entry point n and a unique exit point m so that the entire auxiliary network $\mathcal{G}' = (N', A')$ results from joining the subnetworks $\mathcal{G}'_{nm}$ at the entry and exit points, respectively.

In the auxiliary network we create additional nodes $\tilde{n}_{nm}$ representing potential intermediate states when moving from a node n to a node m. These intermediate nodes model points in time after completion of service at customer n and before start of service at customer m. On the trip between nodes n and m, a driver can drive, take a break or rest, and can take some other off-duty time, e.g., waiting time that is too short to be considered as break or rest. Only after arrival at node m the driver can start the service. If, however, the driver arrives at node m before the opening of the time window, additional break, rest, or waiting periods have to be scheduled before service may begin.

Figure 1 visualizes the subnetwork $\mathcal{G}'_{nm}$ that explicitly models all possible driver activities that can be conducted when moving along an arc $(n, m) \in A$. The six REFs f^{start}_{nm}, $f^{\text{drive}}_{\Delta}$, f^{wait}_{Δ}, f^{rest}_{Δ}, $f^{\text{break}}_{\Delta}$, and f^{service}_{nm} are associated to the arcs as illustrated in the figure. Here, Δ is a parameter of the REFs representing the duration of the respective driver activity.

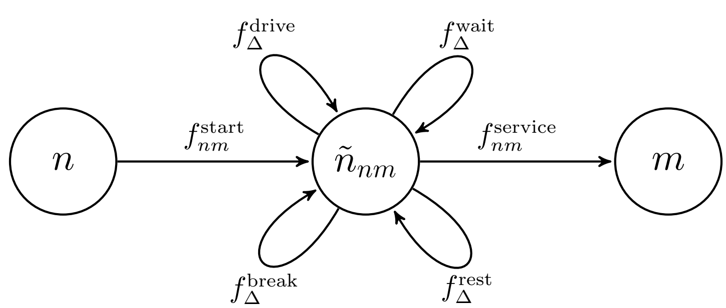

Figure 1 **Subnetwork** $\mathcal{G}'_{nm}$

In order to be able to define these REFs, we first define the necessary resources required for the VRTDSP. A possible driver state at any node can be represented by a tuple

$$l = \left(l^{\text{cost}}, l^{\text{load}}, l^{\text{time}}, l^{\text{dist}}, l^{\text{drive|R}}, l^{\text{elapsed|R}}, l^{\text{elapsed|B}}\right).$$

The semantics of these individual resources is as follows:

l^{cost}: the reduced cost accumulated along the route;

l^{load}: the accumulated load along the route;

l^{time}: the time that has elapsed since the start of the route;

l^{dist}: the remaining driving time to the next customer;

$l^{\text{drive}|\text{R}}$: the accumulated driving time since the last rest;

$l^{\text{elapsed}|\text{R}}$: the time elapsed since the end of the last rest;

$l^{\text{elapsed}|\text{B}}$: the time elapsed since the end of the last break.

In addition to the standard VRPTW resources l^{cost}, l^{time}, and l^{load}, the resource l^{dist} is required because a route can traverse an intermediate node $\tilde{n}_{nm}$ several times before service at node m can begin. Therefore, this resource is relevant only at these intermediate nodes. Furthermore, resources $l^{\text{drive}|\text{R}}$, $l^{\text{elapsed}|\text{R}}$, and $l^{\text{elapsed}|\text{B}}$ are required to ensure compliance with hours of service regulations.

A label representing a fully rested driver at the depot n^{depot} is given by $l := (0, 0, t^{\min}_{n^{\text{depot}}}, 0, 0, 0, 0)$. Given these resource definitions and an initial label, the REFs can be used to update all resource values as required. The REF f^{start}_{nm} updates the accumulated reduced cost and load, but not the time, because the scheduling of driver activities will be done by the other REFs. Moreover, the distance to the next node is initialized to d_{nm} so that the vehicle will later drive for exactly this amount of time. The REFs $f^{\text{drive}}_{\Delta}$, f^{wait}_{Δ}, $f^{\text{break}}_{\Delta}$, and f^{rest}_{Δ} increase the time resource by Δ, i.e., the duration of the driver activity. Furthermore, the remaining distance, the accumulated amount of driving since the last rest, and the time elapsed since the end of the last break and rest are updated appropriately. Similarly, the REF f^{service}_{nm} increases the time resource as well as the time elapsed since the end of the last break and rest by the service time s_m. Table 2 shows in detail how the REFs update the resource values, i.e., when a label l is extended along arc (n, m) to a new label $\hat{l}$. For the sake of simplicity, the blank entries in the table indicate that the resource value is kept

	Resource extension functions (REFs)					
Result $\hat{l}$	$f^{\text{start}}_{nm}(l)$	$f^{\text{drive}}_{\Delta}(l)$	$f^{\text{wait}}_{\Delta}(l)$	$f^{\text{break}}_{\Delta}(l)$	$f^{\text{rest}}_{\Delta}(l)$	$f^{\text{service}}_{nm}(l)$
$\hat{l}^{\text{cost}}$	$l^{\text{cost}} + \tilde{c}_{nm}$					
$\hat{l}^{\text{load}}$	$l^{\text{load}} + q_m$					
$\hat{l}^{\text{time}}$		$l^{\text{time}} + \Delta$	$l^{\text{time}} + \Delta$	$l^{\text{time}} + \Delta$	$l^{\text{time}} + \Delta$	$l^{\text{time}} + s_m$
$\hat{l}^{\text{dist}}$	d_{nm}	$l^{\text{dist}} - \Delta$				
$\hat{l}^{\text{drive}\|\text{R}}$		$l^{\text{drive}\|\text{R}} + \Delta$			0	
$\hat{l}^{\text{elapsed}\|\text{R}}$		$l^{\text{elapsed}\|\text{R}} + \Delta$	$l^{\text{elapsed}\|\text{R}} + \Delta$	$l^{\text{elapsed}\|\text{R}} + \Delta$	0	$l^{\text{elapsed}\|\text{R}} + s_m$
$\hat{l}^{\text{elapsed}\|\text{B}}$		$l^{\text{elapsed}\|\text{B}} + \Delta$	$l^{\text{elapsed}\|\text{B}} + \Delta$	0	0	$l^{\text{elapsed}\|\text{B}} + s_m$

Table 2 **Resource extension functions**

unchanged, e.g., for $\hat{l} := f^{\text{start}}_{nm}(l)$ we have $\hat{l}^{\text{time}} = l^{\text{time}}$. The reduced cost $\tilde{c}_{nm}$ of an arc (n, m) is defined as $c_{nm} - \pi_n$, where π_n is the dual price of the corresponding partitioning constraint (1b) for $n \in C$ and $\pi_{n^{\text{depot}}} = 0$.

Note that the regulation does not impose a general upper limit on $l^{\text{elapsed|R}}$ and $l^{\text{elapsed|B}}$. A label with $l^{\text{elapsed|R}} > t^{\text{elapsed|R}}$ or $l^{\text{elapsed|B}} > t^{\text{elapsed|B}}$ can be feasible, however, if a driving period is scheduled, the respective limits must not be exceeded. Instead of defining resource intervals for all resources, we therefore explicitly state the conditions under which a particular REF creates a feasible label. Given a feasible label l, the label $f^{\text{start}}_{nm}(l)$ is feasible if and only if $l^{\text{load}} + q_m \leq Q$. Furthermore, label $f^{\text{drive}}_{\Delta}(l)$ is feasible if and only if $0 \leq \Delta \leq \Delta_l$ with

$$\Delta_l := \min\{l^{\text{dist}}, t^{\text{drive|R}} - l^{\text{drive|R}}, t^{\text{elapsed|R}} - l^{\text{elapsed|R}}, t^{\text{elapsed|B}} - l^{\text{elapsed|B}}\}, \tag{2}$$

and $f^{\text{wait}}_{\Delta}(l)$ is feasible for all non-negative values of Δ. The labels $f^{\text{break}}_{\Delta}(l)$ and $f^{\text{rest}}_{\Delta}(l)$ are feasible if and only if $\Delta \geq t^{\text{break}}$ and $\Delta \geq t^{\text{rest}}$. Furthermore, label $f^{\text{service}}_{nm}(l)$ is feasible if and only if $t^{\min}_m \leq l^{\text{time}} \leq t^{\max}_m$ and $l^{\text{dist}} = 0$.

Given a route in the network (N, A) and a schedule specifying the type and duration of each driver activity conducted (in general many different schedules are possible), we can find a corresponding walk in the auxiliary network (N', A') and parameter values Δ for each REF applied along the walk that requires such a parameter. Note that (N', A') is a multigraph with parallel arcs. Hence, a unique representation of a walk must also specify the respective REFs with their Δ values (if any). A schedule is feasible if the above mentioned feasibility conditions are satisified for each REF applied along the walk.

Conversely, a walk in the auxiliary network (N', A') which starts and ends at the depot, corresponds to a route in the network (N, A). Together with the respective parameter values Δ, such a walk induces a schedule.

With these definitions of REFs and conditions for feasibility we can determine whether a given schedule is feasible or not. However, determining whether a feasible schedule exists for a given route in the original network, would require to evaluate all possible walks through the auxiliary network with all reasonable parameter values for REFs $f^{\text{drive}}_{\Delta}$, f^{wait}_{Δ}, $f^{\text{break}}_{\Delta}$, and f^{rest}_{Δ}. Such a procedure is certainly not efficient so that an alternative approach is needed.

4.2. A Parameter-Free Model

This section shows that we can replace the auxiliary network $\mathcal{G}'$ by another network $\mathcal{G}''$, in which the respective value of Δ can be uniquely computed before applying a REF. As we will see, we can always use the largest possible value of Δ when applying $f^{\text{drive}}_{\Delta}$, and the smallest possible value of Δ when applying f^{wait}_{Δ}, $f^{\text{break}}_{\Delta}$, and f^{rest}_{Δ}.

By scheduling all off-duty periods as short as possible, we obviously avoid times during which the driver is not productive, however, this may also cause unnecessary waiting time due to time window constraints at subsequent customer locations. Such unproductive waiting time can be avoided if break and rest periods are scheduled with a longer duration than required by the regulation. Anyhow, we can tentatively schedule all rest and break periods with a duration of t^{rest} and t^{break} if we know that we can later extend their duration if this may be beneficial. By extending the duration of a rest or break, the service time at subsequent customer locations may be pushed out of their respective time windows. If we want to be able to extend the duration of rest or break periods, we therefore have to calculate additional resource values indicating the maximum amount by which the duration of a rest or break can be extended without violating constraints. Let $l^{\text{latest}|\text{R}}$ and $l^{\text{latest}|\text{B}}$ be additional resources indicating the latest time at which the last rest and break, respectively, must end such that the service time at any subsequent customer location is not pushed out of the customer's time window. Then, the duration of the latest rest and the latest break can be extended by any value less than or equal to $l^{\text{latest}|\text{R}} - (l^{\text{time}} - l^{\text{elapsed}|\text{R}})$ and $l^{\text{latest}|\text{B}} - (l^{\text{time}} - l^{\text{elapsed}|\text{B}})$, respectively. The new resource values $\hat{l}$ can be calculated from a given label l by the REFs as shown in Table 3. (Again, blank entries indicate that resource values remain unchanged.)

	Resource extension functions (REFs)					
Result $\hat{l}$	$f_{nm}^{\text{start}}(l)$	$f_{\Delta}^{\text{drive}}(l)$	$f_{\Delta}^{\text{wait}}(l)$	$f_{\Delta}^{\text{break}}(l)$	$f_{\Delta}^{\text{rest}}(l)$	$f_{nm}^{\text{service}}(l)$
$\hat{l}^{\text{latest}|\text{R}}$					∞	$\min\{l^{\text{latest}|\text{R}}, t_m^{\max} - l^{\text{elapsed}|\text{R}}\}$
$\hat{l}^{\text{latest}|\text{B}}$		$\min\{l^{\text{latest}|\text{B}}, l^{\text{latest}|\text{R}} + t^{\text{elapsed}|\text{R}} - l^{\text{elapsed}|\text{B}} - \Delta\}$		∞	∞	$\min\{l^{\text{latest}|\text{B}}, t_m^{\max} - l^{\text{elapsed}|\text{B}}\}$

Table 3 Modified resource extension functions

Immediately after applying $f_{\Delta}^{\text{break}}$ or f_{Δ}^{rest}, the duration of the break or rest period can be extended by any value. After applying f_{nm}^{service}, the service period is scheduled and the latest end time of the last break or rest may have to be reduced to guarantee that the service time is not pushed out of the time window. The only intricate case is the update of the resource $l^{\text{latest}|\text{B}}$ in $f_{\Delta}^{\text{drive}}$. If the duration of a break is extended after adding a driving period using $f_{\Delta}^{\text{drive}}$, it must be guaranteed that the end of the driving period is not pushed out of the $t^{\text{elapsed}|\text{R}}$ limit after the end of the last rest. Therefore, $l^{\text{latest}|\text{B}}$ may have to be reduced in such a way that the driving period does not end later than $l^{\text{latest}|\text{R}} + t^{\text{elapsed}|\text{R}}$. Therefore, the last break must not end later than $l^{\text{latest}|\text{R}} + t^{\text{elapsed}|\text{R}} - (l^{\text{elapsed}|\text{B}} + \Delta)$.

Figure 2 illustrates an example of a schedule, in which the driver rests for 10 hours, then drives for 4 hours, loads the vehicle for 2 hours, before driving for another 4 hours. After taking a break of one hour, the corresponding resource values are $l^{\text{drive}|\text{R}} = 8$, $l^{\text{elapsed}|\text{R}} = 11$, $l^{\text{elapsed}|\text{B}} = 0$, $l^{\text{latest}|\text{B}} = \infty$. The propagation with $f_{\Delta}^{\text{drive}}$ is feasible for values $\Delta \leq \min\{11-8, 14-11, 8-0\} = 3$. By adding a driving period of 2 hours duration, the new label $\hat{l} = f_2^{\text{drive}}(l)$ has the resource values $\hat{l}^{\text{drive}|\text{R}} = 10$,

$\hat{l}^{\text{elapsed|R}} = 13$, $\hat{l}^{\text{elapsed|B}} = 2$, and $\hat{l}^{\text{latest|B}} = l^{\text{latest|R}} + 12$. As the break ends 11 hours after the rest, the duration of the last break in this schedule can be extended by at most one hour because otherwise the driver would drive after 14 hours have elapsed since the end of the last rest.

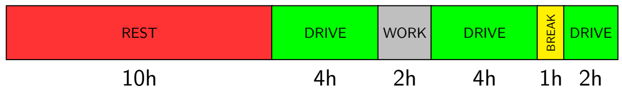

Figure 2 A schedule in which the duration of last break must not be extended by more than one hour.

With the new resource values $l^{\text{latest|R}}$ and $l^{\text{latest|B}}$ we know by how much we can increase the duration of the last rest or break. For a fully rested driver at the depot we can set $l^{\text{latest|R}} := \infty$ and $l^{\text{latest|B}} := \infty$. Let us now define a new REF f_{nm}^{visit}, which not only schedules the service period at customer m, but also any waiting time required during the visit (and before service begins). To reduce this waiting time as much as possible, f_{nm}^{visit} takes advantage of the possibility of increasing the duration of the last rest or break. The new REF f_{nm}^{visit} with $\hat{l} := f_{nm}^{\text{visit}}(l)$ is defined as follows:

$$\begin{aligned}
\hat{l}^{\text{time}} &:= \max\{l^{\text{time}}, t_m^{\min}\} + s_m \\
\hat{l}^{\text{elapsed|R}} &:= \max\{l^{\text{elapsed|R}}, t_m^{\min} - l^{\text{latest|R}}\} + s_m \\
\hat{l}^{\text{latest|R}} &:= \min\{l^{\text{latest|R}}, t_m^{\max} + s_m - \hat{l}^{\text{elapsed|R}}\} \\
\hat{l}^{\text{elapsed|B}} &:= \max\{l^{\text{elapsed|B}}, t_m^{\min} - l^{\text{latest|B}}\} + s_m \\
\hat{l}^{\text{latest|B}} &:= \min\{l^{\text{latest|B}}, t_m^{\max} + s_m - \hat{l}^{\text{elapsed|B}}\}
\end{aligned}$$

(Note that f_{nm}^{visit} leaves all other resource values unchanged.)

Unlike f_{nm}^{service}, the new REF f_{nm}^{visit} does not require that $t_m^{\min} \le l^{\text{time}} \le t_m^{\max}$. If $t_m^{\min} \le l^{\text{time}} \le t_m^{\max}$, then f_{nm}^{service} and f_{nm}^{visit} change the resource tuple in the same way. However, if $l^{\text{time}} < t_m^{\min}$, then f_{nm}^{visit} extends the duration of the last rest and/or the last break in order to avoid unproductive waiting times and adds waiting time if necessary to reach the opening of the time window. Given a feasible label l for a path ending at an intermediate node in the auxiliary network, the label $f_{nm}^{\text{visit}}(l)$ is feasible if and only if $l^{\text{time}} \le t_m^{\max}$ and $l^{\text{dist}} = 0$.

By replacing f_{nm}^{service} with f_{nm}^{visit} in the auxiliary network, we obtain a model in which dominance rules can be formulated allowing us to reduce the number of labels dramatically. Let us write $l_1 \preceq l_2$ if $l_1^i \le l_2^i$ for the resources $i \in \{\text{cost}, \text{time}, \text{load}, \text{dist}, \text{drive|R}, \text{elapsed|R}, \text{elapsed|B}\}$ and $l_1^i \ge l_2^i$ for the resources $i \in \{\text{latest|R}, \text{latest|B}\}$. It is easy to verify that if $l_1 \preceq l_2$, then $f(l_1) \preceq f(l_2)$ for $f \in \{f_{nm}^{\text{start}}, f_\Delta^{\text{drive}}, f_\Delta^{\text{wait}}, f_\Delta^{\text{break}}, f_\Delta^{\text{rest}}, f_{nm}^{\text{visit}}\}$. A direct consequence is that, if $l_1 \preceq l_2$ and $f(l_2)$ is feasible, then $f(l_1)$ is also feasible.

We can now state the following proposition:

PROPOSITION 1. *Let l_1 and l_2 be the resource tuples of different paths P_{l_1} and P_{l_2} ending at the same node in the auxiliary network. If $l_1 \preceq l_2$, then any resource-feasible extension of P_{l_2} is also a resource-feasible extension of P_{l_1} with not greater cost. Hence, l_1 dominates l_2, so that the label l_2 can be discarded in an SPPRC labeling algorithm.*

Proposition 1 has several direct implications. First, because of $l \preceq f^{\text{wait}}_{\Delta}(l)$, we can remove all arcs associated with the REFs f^{wait}_{Δ} from the network $\mathcal{G}'$. Any unavoidable waiting time can be accounted for by REF f^{visit}_{nm}. Such unavoidable waiting occurs if it impossible to extend the duration of the last rest and/or break to such an extend that the opening of the time window is reached, i.e., if $l^{\text{latest|R}} + l^{\text{elapsed|R}} < t^{\min}_m$ or $l^{\text{latest|B}} + l^{\text{elapsed|B}} < t^{\min}_m$ (or both).

Second, we have $f^{\text{rest}}_{t^{\text{rest}}}(l) \preceq f^{\text{rest}}_{\Delta}(l)$ for all $\Delta \geq t^{\text{rest}}$ and $f^{\text{break}}_{t^{\text{break}}}(l) \preceq f^{\text{break}}_{\Delta}(l)$ for all $\Delta \geq t^{\text{break}}$. Therefore, we do not need to evaluate all possible values of Δ when using REFs f^{rest}_{Δ} and $f^{\text{break}}_{\Delta}$. It is sufficient to always use the smallest possible value. The duration of the last rest or break will be extended by REF f^{visit}_{nm} if necessary. Such an extension occurs if the opening of the time window is not yet reached, i.e., $l^{\text{time}} < t^{\min}_m$, and the last rest or break ends before the latest possible point in time, i.e., $(l^{\text{time}} - l^{\text{elapsed|R}}) < l^{\text{latest|R}}$ or $(l^{\text{time}} - l^{\text{elapsed|B}}) < l^{\text{latest|B}}$.

Third, we have

$$f^{\text{rest}}_{t^{\text{rest}}}(l) \preceq f^{\text{rest}}_{t^{\text{rest}}} \circ f^{\text{break}}_{t^{\text{break}}}(l) \tag{3a}$$

$$f^{\text{rest}}_{t^{\text{rest}}}(l) \preceq f^{\text{break}}_{t^{\text{break}}} \circ f^{\text{rest}}_{t^{\text{rest}}}(l) \tag{3b}$$

$$f^{\text{rest}}_{t^{\text{rest}}}(l) \preceq f^{\text{rest}}_{t^{\text{rest}}} \circ f^{\text{rest}}_{t^{\text{rest}}}(l) \tag{3c}$$

$$f^{\text{break}}_{t^{\text{break}}}(l) \preceq f^{\text{break}}_{t^{\text{break}}} \circ f^{\text{break}}_{t^{\text{break}}}(l). \tag{3d}$$

Therefore, it is never beneficial to schedule two consecutive break and/or rest periods and $f^{\text{drive}}_{\Delta_l}$ and either $f^{\text{break}}_{t^{\text{break}}}$ or $f^{\text{rest}}_{t^{\text{rest}}}$ should alternate before applying REF f^{visit}_{nm}.

Fourth, for any given values $\Delta_1, \Delta_2 \geq 0$ for which $f^{\text{drive}}_{\Delta_2} \circ f^{\text{rest}}_{t^{\text{rest}}} \circ f^{\text{drive}}_{\Delta_1}(l)$ or $f^{\text{drive}}_{\Delta_2} \circ f^{\text{break}}_{t^{\text{rest}}} \circ f^{\text{drive}}_{\Delta_1}(l)$ is feasible, we have

$$f^{\text{drive}}_{\Delta_2 - \Delta} \circ f^{\text{rest}}_{t^{\text{rest}}} \circ f^{\text{drive}}_{\Delta_1 + \Delta}(l) \preceq f^{\text{drive}}_{\Delta_2} \circ f^{\text{rest}}_{t^{\text{rest}}} \circ f^{\text{drive}}_{\Delta_1}(l) \tag{4a}$$

$$f^{\text{drive}}_{\Delta_2 - \Delta} \circ f^{\text{break}}_{t^{\text{break}}} \circ f^{\text{drive}}_{\Delta_1 + \Delta}(l) \preceq f^{\text{drive}}_{\Delta_2} \circ f^{\text{break}}_{t^{\text{break}}} \circ f^{\text{drive}}_{\Delta_1}(l) \tag{4b}$$

for any $\Delta \leq \min\{\Delta_l - \Delta_1, \Delta_2\}$, where Δ_l is defined as in (2). Therefore, we can always schedule driving periods with the maximum possible duration (which is Δ_l).

Based on these findings, we can now replace the auxiliary network $\mathcal{G}'_{nm}$ by another network $\mathcal{G}''_{nm}$ depicted in Figure 3. The network $\mathcal{G}''_{nm}$ has two intermediate nodes $\tilde{n}^{\text{fit}}_{nm}$ and $\tilde{n}^{\text{dull}}_{nm}$ for each arc $(n,m) \in A$, instead of only one intermediate node in $\mathcal{G}'_{nm}$. The node $\tilde{n}^{\text{fit}}_{nm}$ represents states, in which

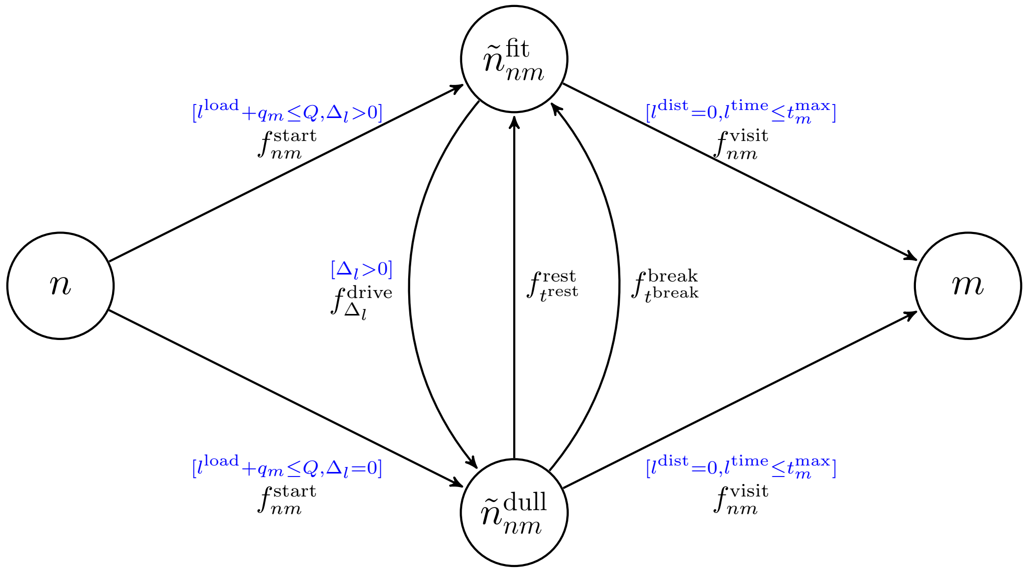

Figure 3 Parameter-free model for efficient labeling

the driver is able to drive or has taken a break or rest after arrival at the next customer location. The node $\tilde{n}^{\text{dull}}_{nm}$ models states, in which the driver requires a break or rest or has just reached the next customer location. Where necessary, preconditions indicating whether a particular arc can be used are shown within square brackets next to the arcs.

Compared to network $\mathcal{G}'$, the main advantage of the network $\mathcal{G}''$ is that all REFs are parameter-free. Furthermore, the network exploits the above findings that $f^{\text{drive}}_{\Delta_l}$ and either $f^{\text{break}}_{t^{\text{break}}}$ or $f^{\text{rest}}_{t^{\text{rest}}}$ should alternate before applying REF f^{visit}_{nm}. With network $\mathcal{G}''$, a standard SPPRC labeling algorithm can be used to solve the pricing problem efficiently. Moreover, all labels at both nodes $\tilde{n}^{\text{fit}}_{nm}$ and $\tilde{n}^{\text{dull}}_{nm}$ can be compared among another so that the dominance conditions of Propositions 1 can be applied on the union of both label sets.

Finally, an additional dominance rule can be given, with which it is possible to further reduce the number of labels that have to be considered.

PROPOSITION 2. *Let l_1 and l_2 be the resource tuples of two different paths P_{l_1} and P_{l_2} ending at the same node in the auxiliary network. If $l^{\text{cost}}_1 \leq l^{\text{cost}}_2$, $l^{\text{load}}_1 \leq l^{\text{load}}_2$, $l^{\text{dist}}_1 \leq l^{\text{dist}}_2$, $l^{\text{time}}_1 + t^{\text{rest}} \leq l^{\text{time}}_2$, and $f^{\text{rest}}_{t^{\text{rest}}}(l_1) \neq l_2$, then the label l_2 can be discarded in an SPPRC labeling algorithm.*

If the labels end at an intermediate node in $N' \setminus N$, then $f^{\text{rest}}_{t^{\text{rest}}}(l_1)$ dominates l_2 because of Proposition 1. Otherwise, for labels ending at an original node in N, a similar argument can be applied to all extensions, in which f^{start}_{nm} is applied first. It follows that $f^{\text{rest}}_{t^{\text{rest}}} \circ f^{\text{start}}_{nm}(l_1) \neq f^{\text{start}}_{nm}(l_2)$ and $f^{\text{rest}}_{t^{\text{rest}}} \circ f^{\text{start}}_{nm}(l_1) \leq f^{\text{start}}_{nm}(l_2)$ for all outgoing arcs of n so that Proposition 1 yields the result also in this case.

4.3. Pricing Elementary Routes

In principle, feasibility of a route regarding capacity, time windows, and hours of service regulations can be modeled and ensured independently from (partial) elementarity constraints. Compliance with k-cycle freeness, the ng-route relaxation or complete elementarity does not impact the resource propagation. For this reason, we have detached issues related to elementarity from the previous sections.

The standard way of imposing elementarity to solve the ESPPRC is to introduce additional binary resources (a.k.a. visiting counters), one for each customer node $n \in C$, to indicate whether or not that customer has already been visited (Beasley and Christofides 1989). For each $i \in C$, an additional resource value $l^{\text{visited}|i}$ can be determined when propagating $\hat{l} := f^{\text{start}}_{nm}(l)$ using

$$\hat{l}^{\text{visited}|i} := l^{\text{visited}|i} + \delta_{i=m}$$

where the *Kronecker delta* is defined as $\delta_{cond} = 1$ if a condition *cond* is true, and $\delta_{cond} = 0$ otherwise. The precondition $l^{\text{visited}|m} = 0$ is added to all arcs in the network $\mathcal{G}''$ associated to REF f^{start}_{nm}. With the modified REF and the updated preconditions only elementary routes are produced via labeling.

In the following we describe several techniques that can be used to accelerate the solution of the ESPPRC, namely, unreachable customers, label loading, limited discrepancy search, and heuristic dominance.

Unreachable Customers For the VRPTW, Feillet et al. (2004) suggested to use resources indicating whether customers have become unreachable instead of visiting counters. A customer $i \in C$ is unreachable for a label l if it is already visited, if the customer demand cannot be satisfied because of capacity constraints, or if the customer cannot be reached before the end of the time window. By using such resources, the dominance rule becomes stronger because more labels become comparable and dominated so that in general less labels need to be extended.

Determining whether a customer is unreachable due to capacity constraints is straight forward. However, determining whether a customer can be reached before the end of the time window is not trivial for the VRTDSP. To precisely determine the set of unreachable nodes for a label l corresponding to a path ending at customer $n \in C$, it would be required to determine for all $(n, m) \in A$ whether a feasible label for m can be calculated. Such a calculation, however, is computationally expensive and should be avoided.

Instead, the set of unreachable nodes can be heuristically determined. Clearly, given a label l for a path ending at customer $n \in C$, a customer $m \in C$ is unreachable if $l^{\text{load}} + q_m > Q$ or $l^{\text{time}} + d_{nm} > t^{\max}_m$. As no breaks and rest periods are considered in this condition, many customers which are unreachable may not be identified.

A better approximation of the set of unreachable nodes in the VRTDSP can be obtained by determining lower bounds on the number of breaks and rest periods along a trip. For U.S. hours of service regulations, we know that the number of compulsory rest periods during a trip along arc $(n,m) \in A$ is at least

$$k_{nm}^{\text{rest}} = \max\left\{0, \left\lceil \frac{d_{nm}}{t^{\text{drive|R}}} - 1 \right\rceil\right\}$$

because a driver must not continue to drive without taking a rest period whenever the maximum amount of accumulated driving between two consecutive rest periods is reached.

In order to determine a lower bound on the number of breaks, we need to consider the interplay between breaks and rests. The most efficient way of driving is to drive 8 hours continuously whenever possible. If k_{nm}^{rest} rest periods are taken, there can be at most $k_{nm}^{\text{rest}}+1$ blocks of 8 hours continuous driving. If $d_{nm} - (k_{nm}^{\text{rest}}+1)t^{\text{elapsed|B}} > 0$, then one or several breaks and driving blocks of up to 3 hours are needed. A lower bound on the number of breaks is

$$k_{nm}^{\text{break}} := \left\lceil \frac{\max\{0, d_{nm} - (k_{nm}^{\text{rest}}+1)t^{\text{elapsed|B}}\}}{t^{\text{drive|R}} - t^{\text{elapsed|B}}} \right\rceil .$$

Figures 4 and 5 show examples illustrating the minimum number of rests and breaks required for different values of d_{nm}.

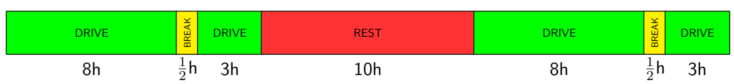

Figure 4 The lower bound on the duration of a trip with 22 hours of driving is 33 hours.

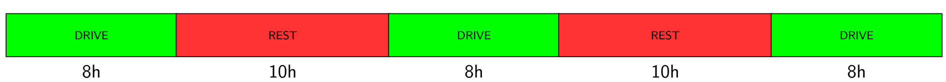

Figure 5 The lower bound on the duration of a trip with 24 hours of driving is 44 hours.

A lower bound on the duration of the trip along arc $(n,m) \in A$ is therefore

$$\hat{d}_{nm} := d_{nm} + k_{nm}^{\text{rest}} t^{\text{rest}} + k_{nm}^{\text{break}} t^{\text{break}} \tag{5}$$

The travel time matrix $(\hat{d}_{nm})_{(n,m)\in A}$ can be easily calculated beforehand in a preprocessing step. For other regulations, lower bound values can be found in a similar fashion.

We can replace the resource values $l^{\text{visited}|i}$ by binary resource values $l^{\text{unreachable}|i}$ indicating whether customer $i \in C$ is identified to be unreachable. For all $i \in C$, the new resource values $\hat{l}^{\text{unreachable}|i}$ are calculated with f_{nm}^{start} and f_{nm}^{visit} as shown in Table 4.

As before, the Kronecker delta is used in the definition of the REFs. The update of $l^{\text{unreachable}|i}$ to account for visited customers is done in REF f_{nm}^{start} and essentially stays the same as before,

Result $\hat{l}$	Resource extension functions (REFs) $f_{nm}^{\text{start}}(l)$	$f_{nm}^{\text{visit}}(l)$
$\hat{l}^{\text{unreachable}\vert i}$	$l^{\text{unreachable}\vert i} + \delta_{i=m}$	$\max\{l^{\text{unreachable}\vert i}, \delta_{l^{\text{load}}+q_i>Q}, \delta_{\max\{l^{\text{time}}, t_m^{\min}\}+s_m+\hat{d}_{mi}>t_i^{\max}}\}$

Table 4 Determination of unreachable customers

except for notational changes. The update of $l^{\text{unreachable}|i}$ to account for customers which cannot be reached due to capacity and time constraints is done in REF f_{nm}^{visit}. For all arcs in the network $\mathcal{G}''$ which are associated to REF f_{nm}^{start}, the precondition $l^{\text{unreachable}|m} = 0$ is used.

Note that we cannot conclude that a customer $i \in C$ is reachable within time windows if $l^{\text{unreachable}|i} = 0$, because we use a lower bound on the arrival time at customer i. Depending on the state of the driver, additional breaks or rests may be required. Thus, we may need to invoke REFs in the subnetwork $\mathcal{G}''_{nm}$ for some labels l even if the customer m is actually not reachable. Anyhow, for any label at an intermediate node in $N' \setminus N$, we can also determine a lower bound on the earliest possible arrival time at the next customer based on l^{dist}, and discard the label if the customer cannot be reached within the time window.

Label Loading As proposed by Feillet et al. (2007), we can increase the performance of the ESPPRC labeling algorithm by generating some high quality labels in a preprocessing step. For any route $r \in R$ with $\lambda_r > 0$, we can a priori generate labels by traversing the route. Since each of these routes has reduced costs of zero, these labels are likely to dominate several other labels that are later generated when solving the ESPPRC.

Limited Discrepancy Search A common approach for accelerating column-generation algorithms is to start by solving the pricing problem heuristically. The exact ESPPRC labeling algorithm is then invoked only when the heuristic fails. Feillet et al. (2007) proposed to use *limited discrepancy search* (LDS) to help the search finding columns with negative reduced costs quickly by only considering the most promising labels. When extending a label l resident at node n, the set of outgoing arcs (n,m) is partitioned into two sets: the set of *good arcs* including the arcs with the lowest cost and the return arc to the depot, and the set of *bad arcs* which includes all other arcs. An additional attribute l^{bad} is added to the labels in order to record the number of bad arcs in the path. Every time a bad arc is traversed, the value l^{bad} is incremented. Within LDS only labels with $l^{\text{bad}} \leq \Lambda$ are considered, where Λ is a parameter called the *discrepancy limit.* LDS starts with a discrepancy limit of $\Lambda = 0$ and solves the ESPPRC. If no path with negative reduced cost is found, the discrepancy limit is increased and the ESPPRC is resolved with the new limit. In the beginning of the search, the discrepancy value is increased by one in each iteration. After a few iterations, the discrepancy limit is set to a sufficiently high value to guarantee that the ESPPRC is solved to optimality. By this, time consuming iterations with high discrepancy limits are avoided.

Heuristic Dominance In order to further increase the effectiveness of solving the pricing problem, we can exploit the special structure of the VRTDSP. Recall that several different labels may exist for the same path in the original network $\mathcal{G}$. They result from different schedules, i.e., varying paths in the network $\mathcal{G}''$. Let P denote a path in the original network $\mathcal{G}$ and let $\mathcal{L}(P)$ denote the set of labels belonging to this path. For each label $l \in \mathcal{L}(P)$ we know that l^{cost}, l^{load}, and l^{bad} are identical. Let $l_P \in \mathcal{L}(P)$ be a label with $l_P^{\text{time}} \leq l^{\text{time}}$ for all $l \in \mathcal{L}(P)$.

For all non-dominated labels $l \in \mathcal{L}(P)$, we have $l_P^{\text{time}} \leq l^{\text{time}} \leq l_P^{\text{time}} + t^{\text{rest}}$ because of Proposition 2. Furthermore, $l_P^{\text{unreachable}|i} \leq l^{\text{unreachable}|i}$ holds for all $i \in C$.

For any path P and any label $l \in \mathcal{L}(P)$, our heuristic dominance rule compares labels only on the basis of the resources l^{cost}, l^{load}, l^{time}, l^{dist}, l^{bad}, and $l_P^{\text{unreachable}|i}$. The remaining resources are disregarded. The advantage of using this overly strict dominance rule is that more labels are dominated so that less labels need to be processed. On the downside, some labels may be discarded although they would provide Pareto-optimal or even minimal-cost paths. If pricing with the stricter dominance fails, we resolve with the exact dominance rule. In particular for more difficult instances, the redundant calculations due to failures are justified by the improved speed gained by the heuristic dominance rule.

4.4. Possible Extensions

Our implementation of branch-and-price has been developed with the goal of demonstrating that it is possible to optimally solve vehicle routing problems in which complex driver rules have to be considered. Compared to the most recent VRPTW branch-and-price implementations, it certainly leaves enough room for possible improvements. First, the incorporation of valid inequalities could help to further tighten the RMP bound. Examples are the (extended) k-path cuts (Kohl et al. 1999, Desaulniers et al. 2008) and subset row inequalities (Jepsen et al. 2008). Second, instead of solving the elementary pricing problem, the ng-route relaxations introduced by Baldacci et al. (2011a) often provide very tight bounds, but are computationally less costly. Bounding techniques can substantially accelerate the labeling process (Baldacci et al. 2011a, Bode and Irnich 2014) in combination with bidirectional labeling (Righini and Salani 2006), while the latter is certainly non-trivial for the VRTDSP. The careful selection of customers to include in the respective neighborhoods is discussed and analyzed by Roberti and Mingozzi (2014) and Bode and Irnich (2014). Third, more sophisticated pricing heuristics can be used. Pricing heuristics may be based on network reduction techniques, gradually modified stronger dominance rules, and metaheuristics such as tabu search (Desaulniers et al. 2008). Even if all these powerful techniques are nice to have, they are far beyond the scope of the paper at hand.

4.5. Adaptations for Other Regulations

The presentation of the approach so far aimed at hours of service regulations in the United States. The general framework of this approach can also be used for other regulations. However, as other regulations may have different definitions of driver activities, the auxiliary model presented in Section 4.1 may have a different number of arcs leaving and entering node $\tilde{n}_{nm}$ with differently defined REFs and resource labels. The main challenge in adapting our approach for other regulations is the generation of a parameter-free model similar to the one presented in Section 4.2. If such a model can be developed in a way that the resulting number of non-dominated alternative labels remains reasonable, the presented algorithmic framework can be used for the regulation.

In the following, we briefly describe the changes required for the VRTDSP in the European Union focusing on the same set of rules considered in Goel (2010). These rules guarantee compliance with European law. However, further rules allowing for more flexibility exist and each member state of the European Union has additional national regulations which cannot be considered in this paper. The basic set of rules in the European Union requires that truck drivers take a break or rest after at most four and a half hours of driving and a rest after at most nine hours of driving. A break must have a duration of 45 minutes, whereas a rest must be an uninterrupted period of at least 11 hours duration. The required rest must be taken within 24 hours after the end of the previous rest. Furthermore, drivers may take breaks and rest periods in two parts. If they do so, the first part of the break must have a duration of at least 15 minutes and the second part of at least 30 minutes, whereas the first part of the rest must have a duration of at least three hours and the second part of at least nine hours.

The basic rules without the option to take breaks and rest periods in two parts can be modeled using the same resource values, REFs, and parameter-free network presented for U.S. regulations except that instead of resource values $l^{\text{elapsed|B}}$ and $l^{\text{latest|B}}$ we only need one resource value $l^{\text{drive|B}}$ indicating the accumulated driving time since the end of the last break or rest. This value must not exceed $t^{\text{drive|B}} = 4\frac{1}{2}$ hours. As there is no benefit in increasing the duration of a break period to above $t^{\text{break}} = \frac{3}{4}$ hours, the respective adaptation of the REFs is simple. The calculation of the maximum amount of additional driving time is changed to

$$\Delta_l := \min\left\{l^{\text{dist}}, t^{\text{drive|R}} - l^{\text{drive|R}}, t^{\text{drive|B}} - l^{\text{drive|B}}, t^{\text{day}} - (l^{\text{elapsed|R}} + t^{\text{rest}})\right\},$$

where t^{day} represents the duration of a day, i.e. 24 hours. The lower bound on the number of compulsory rest periods can be computed in the same way as for U.S. regulations, i.e. $k^{\text{rest}}_{nm} = \max\left\{0, \left\lceil \frac{d_{nm}}{t^{\text{drive|R}}} - 1 \right\rceil\right\}$, whereas the lower bound on the number of compulsory break periods is $k^{\text{break}}_{nm} = \max\left\{0, \left\lceil \frac{d_{nm}}{t^{\text{drive|B}}} - 1 \right\rceil - k^{\text{rest}}_{nm}\right\}$. An additional feasibility condition for all REFs f is that

$\big(f(l)\big)^{\text{elapsed|R}} + t^{\text{rest}} \leq t^{\text{day}}$ must hold because otherwise it would not be possible to take the next rest within 24 hours after the end of the previous rest. In the Appendix, resource values, REFs, and a parameter-free model are provided that can be used for the case of considering the possibility of taking breaks and rests in two parts.

For U.S. hours of service regulations and the European rules stated above, a parameter-free model can be developed because it is always better to increase the duration of the last rest period instead of adding waiting time to the end of the schedule. For hours of service regulations in Australia or Canada, however, this property does not hold and, so far, it is unknown whether a parameter-free model can be developed. Previous work on these regulations (Goel et al. 2012, Goel and Rousseau 2012) relied on time discretization and partial enumeration to determine whether a feasible truck driver schedule exists for a given sequence of customers. Thus, many thousands of alternative labels must be generated and solving the pricing problem would become extremely time consuming.

5. Computational Results

To evaluate the performance of the proposed algorithm we tested the approach using the 56 benchmark instances for the VRTDSP proposed by Goel (2009) which can be obtained from `http://www.telematique.eu/research/downloads`. These instances are derived from the well-known VRPTW benchmark instances of Solomon (1987) and are grouped into six classes. In classes R1 and R2 customers are randomly distributed in a square region. In classes C1 and C2 customers are clustered, and in classes RC1 and RC2 the customer distribution is mixed. The average size of time windows per instance ranges from less than 7 hours to more than 107 hours. The service time at every customer is set to one hour. The planning horizon is 144 hours and the maximum driving time (without compulsory breaks and rests) required to go from one point in the square region to another is approximately one day. Each instance contains 100 customers, but smaller instances are created considering only the first 25 or 50 customers.

The algorithm is implemented in C++ and CPLEX 12.6.1 is used for solving the restricted master problem. Experiments are run on a single core of an Intel(R) Core(TM)2 Quad CPU Q9300 @ 2.50 GHz processor with a run time limit of 2 hours.

The initial solution is obtained using an adaptation of the savings algorithm proposed by Clarke and Wright (1964). Solving the ESPPRC is prematurely stopped as soon as 500 paths with negative reduced cost are found. The maximum discrepancy limit before solving the ESPPRC to optimality is set to 3. All integer solutions obtained while solving the restricted master problem are used to update the upper bound. If the root node has a fractional solution, we use the standard MIP solver of CPLEX to obtain a good upper bound before starting to branch. We use the standard branching on individual arcs $(n,m) \in A$ of the original network in order to finally enforce integer solutions in the RMP. The branch that forbids an arc $(n,m) \in A$ is implemented by deleting arc $(n,\tilde{n}_{nm})$ from the network $\mathcal{G}''$. The other branch that enforces an arc (n,m), is implemented by deleting arcs $(n,\tilde{n}_{ni})$ with $i \neq m$ from the network $\mathcal{G}''$. Furthermore, the nodes of the branch-and-bound tree are selected in a best-first order with respect to the solution value of the linear relaxation.

We tested our approach for hours of service regulations in the United States as well as European Union regulations and also evaluated the heuristic dominance proposed in Section 4.3.

5.1. U.S. hours of service regulations

Tables 5 and 6 show the results of our experiments for the old and the new regulations in the United States and instances with 25 and 50 customers. In order to consider the old rules we could have simply set $t_{\text{elapsed}|\text{B}}$ to a sufficiently large value, however, we decided to avoid computational overhead by removing all resources and steps of the algorithm related to the break provision. The tables show the solution value of the linear relaxation (LB^{LP}), the best integer solution found before starting the branch-and-bound procedure (UB^{LP}), and the time (in seconds) required to solve the linear relaxation (CPU^{LP}). Furthermore, the lower bound value (LB^{IP}) and the best integer solution found (UB^{IP}) of the the branch-and-bound are shown as well as the overall running time required to solve the instance (CPU^{IP}). Numbers shown in italics are used if the linear relaxation or the overall problem are not solved to optimality.

As can be seen, our approach solves all 56 instances with 25 customers for the old and for 55 instances for the new regulations within the run time limit. It must be noted, that the approach actually finds the optimal solution for all instances, however, without being able to prove optimality for instance TDS_RC208 for the new regulations which was solved to optimality within 3 hours. For instances with 50 customers, the approach finds optimal solutions for 34 and 30 instances for the old and the new regulations.

Table 7 shows a comparison of the average distance of the best integer solutions found by our approach for 100 customer instances with the results of a tabu search algorithm presented by Rancourt et al. (2013). As Rancourt et al. (2013) only consider the old regulations, no comparison for the new regulations can be made. Here, our approach produces better average solution values for two of the six classes of instances. Overall, the average cost using our approach is 3.4 % above the values presented by Rancourt et al. (2013). For both the new and the old regulations our approach was able to optimally solve six instances within the runtime limit. For these six instances, an integer solution was found for the linear relaxation so that no branching was required.

5.2. EU hours of service regulations

Table 8 show the results of our experiments for all EU rules described in Section 4.5 and instances with 25 and 50 customers. The table shows that our approach solves 53 of 56 instances with 25

	U.S. regulations (before 2013)						**U.S. regulations (after 2013)**					
	Linear relaxation			**Integer**			**Linear relaxation**			**Integer**		
Instance	**LB**^{LP}	**UB**^{LP}	**CPU**^{LP}	**LB**^{IP}	**UB**^{IP}	**CPU**^{IP}	**LB**^{LP}	**UB**^{LP}	**CPU**^{LP}	**LB**^{IP}	**UB**^{IP}	**CPU**^{IP}
TDS_C101	191.17	191.17	0.9	191.17	191.17	0.9	191.17	191.17	1.0	191.17	191.17	1.0
TDS_C102	190.08	190.08	5.3	190.08	190.08	5.3	190.08	190.08	5.6	190.08	190.08	5.6
TDS_C103	189.42	189.42	10.7	189.42	189.42	10.7	189.42	189.42	38.6	189.42	189.42	38.6
TDS_C104	186.46	188.17	1619.4	186.67	186.67	2943.1	186.46	188.58	691.9	186.67	186.67	4896.5
TDS_C105	191.17	191.17	1.1	191.17	191.17	1.2	191.17	191.17	1.5	191.17	191.17	1.5
TDS_C106	191.17	191.17	1.0	191.17	191.17	1.1	191.17	191.17	1.2	191.17	191.17	1.2
TDS_C107	191.17	191.17	2.3	191.17	191.17	2.3	191.17	191.17	4.3	191.17	191.17	4.4
TDS_C108	189.75	189.75	9.9	189.75	189.75	9.9	189.75	189.75	5.9	189.75	189.75	5.9
TDS_C109	187.63	187.83	25.5	187.83	187.83	66.7	187.63	191.17	47.3	187.83	187.83	148.1
TDS_C201	226.49	253.75	2.4	248.00	248.00	367.6	226.49	253.75	3.9	248.00	248.00	409.0
TDS_C202	217.83	217.83	10.4	217.83	217.83	10.4	217.83	217.83	14.3	217.83	217.83	14.3
TDS_C203	217.83	217.83	29.0	217.83	217.83	29.0	217.83	217.83	22.1	217.83	217.83	22.2
TDS_C204	214.17	214.17	57.9	214.17	214.17	58.0	214.17	214.17	175.0	214.17	214.17	175.1
TDS_C205	214.42	214.42	1.8	214.42	214.42	1.8	214.42	214.42	2.2	214.42	214.42	2.3
TDS_C206	214.42	214.42	7.6	214.42	214.42	7.7	214.42	214.42	9.3	214.42	214.42	9.3
TDS_C207	214.17	214.17	25.3	214.17	214.17	25.3	214.17	214.17	21.0	214.17	214.17	21.0
TDS_C208	214.17	214.17	13.9	214.17	214.17	13.9	214.17	214.17	9.8	214.17	214.17	9.9
TDS_R101	495.92	502.25	0.5	502.25	502.25	1.6	495.92	503.50	0.8	502.25	502.25	2.3
TDS_R102	446.25	446.25	1.2	446.25	446.25	1.2	446.25	446.25	1.1	446.25	446.25	1.2
TDS_R103	401.08	401.08	1.4	401.08	401.08	1.5	401.08	401.08	2.5	401.08	401.08	2.5
TDS_R104	359.42	359.42	4.6	359.42	359.42	4.6	359.42	359.42	4.9	359.42	359.42	5.0
TDS_R105	435.04	438.17	1.4	438.17	438.17	2.0	435.04	438.17	1.9	438.17	438.17	2.5
TDS_R106	407.08	407.08	1.4	407.08	407.08	1.4	407.08	407.08	2.3	407.08	407.08	2.4
TDS_R107	386.88	400.58	2.8	391.83	391.83	15.0	386.88	391.83	4.6	391.83	391.83	27.1
TDS_R108	345.72	363.58	20.9	349.42	349.42	148.7	345.72	351.00	22.5	349.42	349.42	153.1
TDS_R109	371.92	390.67	3.7	383.33	383.33	23.6	376.13	389.25	3.9	385.08	385.08	40.7
TDS_R110	349.53	359.67	5.6	354.42	354.42	29.5	350.45	359.67	12.9	354.42	354.42	78.0
TDS_R111	379.92	390.58	3.5	387.67	387.67	18.5	384.67	387.67	4.8	387.67	387.67	28.7
TDS_R112	330.73	337.33	95.6	337.33	337.33	472.6	331.67	349.75	80.9	337.33	337.33	1149.8
TDS_R201	460.46	463.58	0.8	463.58	463.58	1.2	460.46	477.83	1.5	463.58	463.58	2.1
TDS_R202	410.75	410.75	2.5	410.75	410.75	2.5	410.75	410.75	2.1	410.75	410.75	2.1
TDS_R203	391.83	391.83	3.3	391.83	391.83	3.3	391.83	391.83	4.7	391.83	391.83	4.7
TDS_R204	352.83	358.25	8.5	355.17	355.17	38.2	353.21	358.25	14.6	355.17	355.17	47.6
TDS_R205	399.69	408.58	3.1	403.67	403.67	7.6	402.73	406.50	3.6	404.08	404.08	7.5
TDS_R206	373.96	386.17	3.8	375.25	375.25	12.6	376.10	380.33	6.2	378.08	378.08	55.6
TDS_R207	361.92	361.92	5.5	361.92	361.92	5.5	367.17	367.17	9.9	367.17	367.17	10.0
TDS_R208	333.21	335.00	30.1	335.00	335.00	156.9	339.25	361.83	42.7	341.08	341.08	251.4
TDS_R209	369.67	378.42	3.4	376.75	376.75	10.2	369.67	376.75	4.8	376.75	376.75	19.3
TDS_R210	407.75	407.75	3.2	407.75	407.75	3.2	408.56	414.58	4.3	411.75	411.75	19.8
TDS_R211	350.72	351.17	7.5	351.17	351.17	28.3	351.17	351.17	11.1	351.17	351.17	11.2
TDS_RC101	358.25	358.25	1.3	358.25	358.25	1.3	358.25	358.25	1.5	358.25	358.25	1.5
TDS_RC102	335.92	335.92	3.3	335.92	335.92	3.4	335.92	335.92	4.3	335.92	335.92	4.4
TDS_RC103	327.08	327.08	8.7	327.08	327.08	8.7	327.08	327.08	6.8	327.08	327.08	6.8
TDS_RC104	299.75	299.75	25.6	299.75	299.75	25.6	299.75	299.75	62.9	299.75	299.75	63.1
TDS_RC105	334.75	334.75	2.3	334.75	334.75	2.3	334.75	334.75	4.0	334.75	334.75	4.1
TDS_RC106	310.83	310.83	3.3	310.83	310.83	3.3	310.83	310.83	4.2	310.83	310.83	4.2
TDS_RC107	296.33	296.33	6.3	296.33	296.33	6.4	296.33	296.33	18.8	296.33	296.33	18.9
TDS_RC108	294.50	294.50	99.4	294.50	294.50	99.5	294.50	294.50	436.5	294.50	294.50	436.6
TDS_RC201	360.50	360.50	0.8	360.50	360.50	0.8	360.50	360.50	0.9	360.50	360.50	1.0
TDS_RC202	338.17	338.17	4.6	338.17	338.17	4.6	338.17	338.17	3.0	338.17	338.17	3.0
TDS_RC203	327.08	327.08	4.0	327.08	327.08	4.0	327.08	327.08	5.7	327.08	327.08	5.8
TDS_RC204	299.75	299.75	9.4	299.75	299.75	9.5	299.75	299.75	11.8	299.75	299.75	11.9
TDS_RC205	338.08	338.08	2.2	338.08	338.08	2.2	338.08	338.08	2.9	338.08	338.08	2.9
TDS_RC206	324.25	324.25	2.1	324.25	324.25	2.1	324.25	324.25	3.6	324.25	324.25	3.7
TDS_RC207	298.33	298.33	2.1	298.33	298.33	2.1	298.33	298.33	2.8	298.33	298.33	2.9
TDS_RC208	289.83	294.50	116.7	294.33	294.33	6007.5	290.42	294.50	183.9	*293.77*	*294.50*	*7200.0*
Average			41.6			191.6			36.7			276.1

Table 5 Results for instances with 25 customers

	U.S. regulations (before 2013)						**U.S. regulations (after 2013)**					
	Linear relaxation			**Integer**			**Linear relaxation**			**Integer**		
Instance	**LB**^{LP}	**UB**^{LP}	**CPU**^{LP}	**LB**^{IP}	**UB**^{IP}	**CPU**^{IP}	**LB**^{LP}	**UB**^{LP}	**CPU**^{LP}	**LB**^{IP}	**UB**^{IP}	**CPU**^{IP}
TDS_C101	362.17	362.17	9.6	362.17	362.17	9.7	362.17	362.17	9.0	362.17	362.17	9.3
TDS_C102	361.08	361.08	25.1	361.08	361.08	25.3	361.08	361.08	23.8	361.08	361.08	24.2
TDS_C103	360.42	360.42	219.9	360.42	360.42	220.1	360.42	360.42	230.3	360.42	360.42	230.8
TDS_C104		*358.92*	*7200.0*					*358.92*	*7200.0*			
TDS_C105	362.17	362.17	14.5	362.17	362.17	14.6	362.17	362.17	21.7	362.17	362.17	21.9
TDS_C106	362.17	362.17	15.7	362.17	362.17	15.8	362.17	362.17	13.2	362.17	362.17	13.4
TDS_C107	362.17	362.17	23.5	362.17	362.17	23.7	362.17	362.17	30.2	362.17	362.17	30.5
TDS_C108	360.75	360.75	43.4	360.75	360.75	43.5	360.75	360.75	50.5	360.75	360.75	50.9
TDS_C109	358.63	360.00	262.8	358.83	358.83	1792.7	358.63	362.17	326.8	358.83	358.83	3157.2
TDS_C201	383.78	498.25	1464.2	*407.97*	*498.25*	*7200.0*	383.78	510.75	370.1	*407.93*	*510.75*	*7200.0*
TDS_C202	370.98	394.25	1016.9	*373.82*	*394.25*	*7200.0*	371.19	399.08	376.2	*373.60*	*399.08*	*7200.0*
TDS_C203		*362.75*	*7200.0*					*364.33*	*7200.0*			
TDS_C204		*366.67*	*7200.0*					*366.42*	*7200.0*			
TDS_C205	364.38	371.17	136.9	369.42	369.42	3823.7	364.38	374.75	192.8	369.42	369.42	4683.2
TDS_C206	364.38	376.92	160.7	369.42	369.42	7016.5	364.38	376.92	226.7	*369.21*	*376.92*	*7200.0*
TDS_C207	362.94	368.67	665.2	*364.04*	*368.67*	*7200.0*	362.94	368.67	777.6	*364.16*	*368.67*	*7200.0*
TDS_C208	363.44	370.92	201.9	*368.11*	*369.17*	*7200.0*	363.44	369.17	318.2	*367.08*	*369.17*	*7200.0*
TDS_R101	847.83	847.83	3.4	847.83	847.83	3.5	847.83	847.83	5.7	847.83	847.83	5.9
TDS_R102	750.42	757.75	17.7	753.92	753.92	118.5	750.42	753.92	21.9	753.92	753.92	205.8
TDS_R103	641.71	654.17	36.8	649.08	649.08	4305.5	642.10	666.67	54.7	649.08	649.08	5186.5
TDS_R104		*538.33*	*7200.0*					*545.83*	*7200.0*			
TDS_R105	738.38	743.92	8.9	743.92	743.92	131.1	743.29	754.25	18.2	749.58	749.58	509.6
TDS_R106	677.18	678.00	48.6	678.00	678.00	113.5	679.93	693.67	47.5	*687.46*	*687.67*	*7200.0*
TDS_R107	596.91	617.50	391.3	*599.62*	*617.50*	*7200.0*	601.98	616.50	495.6	*605.71*	*616.50*	*7200.0*
TDS_R108		*555.67*	*7200.0*					*548.00*	*7200.0*			
TDS_R109	624.26	632.67	77.7	632.25	632.25	2354.2	625.78	656.67	93.3	*636.26*	*638.50*	*7200.0*
TDS_R110	566.75	566.75	554.1	566.75	566.75	554.4	569.08	579.33	483.3	*570.96*	*579.33*	*7200.0*
TDS_R111	575.16	603.67	360.9	*580.90*	*603.67*	*7200.0*	576.16	603.67	390.6	*580.34*	*603.67*	*7200.0*
TDS_R112		*541.67*	*7200.0*					*541.67*	*7200.0*			
TDS_R201	797.50	801.17	10.1	798.92	798.92	31.0	797.50	801.17	14.8	798.92	798.92	35.5
TDS_R202	708.61	714.33	30.1	710.58	710.58	192.0	712.31	722.50	36.2	714.33	714.33	469.2
TDS_R203	612.82	633.83	105.9	*619.62*	*633.83*	*7200.0*	615.79	653.83	219.6	*620.11*	*653.83*	*7200.0*
TDS_R204		*537.92*	*7200.0*					*592.00*	*7200.0*			
TDS_R205	690.53	701.67	22.3	695.25	695.25	242.5	691.38	699.33	32.7	695.25	695.25	263.0
TDS_R206	634.75	656.17	110.4	*640.14*	*656.17*	*7200.0*	634.91	651.83	127.3	*639.77*	*651.83*	*7200.0*
TDS_R207	569.90	606.67	928.7	*571.76*	*606.67*	*7200.0*	571.42	598.33	1350.3	*572.90*	*598.33*	*7200.0*
TDS_R208		*553.58*	*7200.0*					*561.08*	*7200.0*			
TDS_R209	609.65	631.17	73.4	615.58	615.58	1107.5	610.89	634.92	89.6	615.58	615.58	886.5
TDS_R210	645.27	689.92	134.6	*654.21*	*662.83*	*7200.0*	646.59	690.25	149.1	*654.15*	*666.50*	*7200.0*
TDS_R211	547.12	574.00	1296.3	*548.98*	*574.00*	*7200.0*	548.17	570.25	1477.4	*549.77*	*570.25*	*7200.0*
TDS_RC101	632.58	632.58	7.9	632.58	632.58	8.0	632.58	632.58	8.9	632.58	632.58	9.2
TDS_RC102	602.25	602.25	24.0	602.25	602.25	24.2	604.42	604.42	42.8	604.42	604.42	43.4
TDS_RC103	584.67	584.67	628.5	584.67	584.67	628.7	584.67	584.67	82.6	584.67	584.67	83.3
TDS_RC104		*522.92*	*7200.0*					*522.92*	*7200.0*			
TDS_RC105	613.75	613.75	18.7	613.75	613.75	18.9	613.75	613.75	30.3	613.75	613.75	30.8
TDS_RC106	564.92	564.92	59.0	564.92	564.92	59.2	564.92	564.92	88.5	564.92	564.92	89.0
TDS_RC107	522.67	522.67	600.6	522.67	522.67	600.8	522.67	522.67	655.7	522.67	522.67	656.5
TDS_RC108	517.67	517.67	5227.1	517.67	517.67	5227.4	517.67	517.67	6672.3	517.67	517.67	6673.3
TDS_RC201	684.83	684.83	6.1	684.83	684.83	6.2	684.83	684.83	10.5	684.83	684.83	10.8
TDS_RC202	613.83	613.83	25.7	613.83	613.83	25.9	613.83	613.83	21.3	613.83	613.83	21.9
TDS_RC203	594.92	594.92	31.2	594.92	594.92	31.3	594.92	594.92	56.8	594.92	594.92	57.6
TDS_RC204	486.75	489.25	6270.8	*486.75*	*489.25*	*7200.0*		*493.83*	*7200.0*			
TDS_RC205	631.83	631.83	17.4	631.83	631.83	17.5	631.83	631.83	18.9	631.83	631.83	19.3
TDS_RC206	610.17	610.17	20.8	610.17	610.17	20.9	610.17	610.17	22.3	610.17	610.17	22.7
TDS_RC207	560.00	560.00	74.7	560.00	560.00	74.9	560.00	560.00	105.1	560.00	560.00	105.8
TDS_RC208		*517.67*	*7200.0*					*518.50*	*7200.0*			
Average			1669.4						1698.0			

Table 6 **Results for instances with 50 customers**

	U.S. regulations (before 2013)	
	Column generation	Tabu search
C1	825.61	827.29
C2	750.33	644.75
R1	1038.37	985.33
R2	1003.82	967.59
RC1	1166.47	1128.78
RC2	1098.82	1138.96
Average	980.57	948.78

Table 7 Comparison of best integer solutions

customers and 18 of 56 instances with 50 customers. Without run time limit we were able to optimally solve the remaining 25 customer instances TDS_C104, TDS_C109, and TDS_RC208 with solution values of 186.67, 187.83, and 293.50. Although the time required to solve instances for EU regulations is higher than for U.S. regulations, it appears that the increase of the computational effort is still reasonable.

5.3. Heuristic dominance

Most of the acceleration techniques presented in Section 4.3 are adaptations of previously developed techniques for solving the VRPTW, and their impact on the efficiency of the exact approach should be similar for the VRTDSP. To evaluate the impact of the newly proposed *heuristic dominance* for the VRTDSP, we conducted additional experiments in which we used our approach without heuristic dominance. Table 9 shows the average of the time required without using heuristic dominance divided by time required when using heuristic dominance. The average is based on those instances in which both variants of our approach were able to solve the instance within the runtime limit of 2 hours. It can be seen that the heuristic dominance is useful for all regulations tested and for the new U.S. regulations a speed up factor of around 1.5 and for EU regulations a speed up factor of above five was obtained. This indicates that the positive impact of using the heuristic dominance increases for more complex regulations.

6. Conclusions

Despite the importance for many real-life applications, research on solving vehicle routing and truck driver scheduling problems (VRTDSP) is still in its infancy and so far only heuristic approaches had been proposed. In this paper it is shown that the VRTDSP can be solved to proven optimality for U.S. hours of service regulations and EU regulations. We proposed a branch-and-price algorithm, which employs a powerful dynamic programming-based labeling algorithm for the generation of routes complying with the regulations. The power of the approach can be attributed to careful choice and definition of resources together with their resource extension functions (REFs). These allowed us to define an extended network for the shortest path computation, in which REFs are

	25 customers						**50 customers**					
	Linear relaxation			**Integer**			**Linear relaxation**			**Integer**		
Instance	$\mathbf{LB}^{LP}$	$\mathbf{UB}^{LP}$	$\mathbf{CPU}^{LP}$	$\mathbf{LB}^{IP}$	$\mathbf{UB}^{IP}$	$\mathbf{CPU}^{IP}$	$\mathbf{LB}^{LP}$	$\mathbf{UB}^{LP}$	$\mathbf{CPU}^{LP}$	$\mathbf{LB}^{IP}$	$\mathbf{UB}^{IP}$	$\mathbf{CPU}^{IP}$
TDS_C101	191.17	191.17	30.3	191.17	191.17	30.6	362.17	362.17	505.6	362.17	362.17	508.0
TDS_C102	190.08	190.08	271.1	190.08	190.08	271.4	361.08	361.08	1230.7	361.08	361.08	1233.1
TDS_C103	189.42	189.42	363.1	189.42	189.42	363.4		360.42	7200.0			
TDS_C104		188.58	7200.0					358.92	7200.0			
TDS_C105	191.17	191.17	224.8	191.17	191.17	225.0	362.17	362.17	3178.6	362.17	362.17	3181.1
TDS_C106	191.17	191.17	76.9	191.17	191.17	77.2	362.17	362.17	1876.0	362.17	362.17	1878.6
TDS_C107	191.17	191.17	474.8	191.17	191.17	475.1	362.17	362.17	4324.1	362.17	362.17	4327.2
TDS_C108	189.75	189.75	1241.4	189.75	189.75	1241.8		360.75	7200.0			
TDS_C109	187.63	190.58	3531.8	187.63	190.58	7200.0		362.17	7200.0			
TDS_C201	223.17	242.75	296.2	230.58	230.58	3600.5		450.92	7200.0			
TDS_C202	217.83	217.83	287.9	217.83	217.83	288.3		403.42	7200.0			
TDS_C203	217.83	217.83	354.3	217.83	217.83	354.7		372.42	7200.0			
TDS_C204	217.58	217.58	6823.4	217.58	217.58	6823.8		380.42	7200.0			
TDS_C205	214.42	214.42	323.9	214.42	214.42	324.4		462.17	7200.0			
TDS_C206	214.42	214.42	424.8	214.42	214.42	425.2		385.67	7200.0			
TDS_C207	214.17	214.17	659.0	214.17	214.17	659.5		376.00	7200.0			
TDS_C208	214.17	214.17	763.3	214.17	214.17	763.7		378.08	7200.0			
TDS_R101	500.50	506.83	6.1	506.83	506.83	52.0	857.75	866.00	172.6	862.58	862.58	6831.3
TDS_R102	446.92	446.92	11.4	446.92	446.92	11.7	759.71	759.92	1284.0	759.75	759.92	7200.0
TDS_R103	405.39	410.17	16.4	408.00	408.00	365.5	643.81	657.08	1865.9	644.59	657.08	7200.0
TDS_R104	359.42	359.42	41.6	359.42	359.42	41.9		560.58	7200.0			
TDS_R105	438.17	438.17	42.8	438.17	438.17	43.2	743.67	753.67	4162.3	743.67	753.67	7200.0
TDS_R106	407.08	407.08	39.4	407.08	407.08	39.8	680.62	694.42	1715.0	680.62	694.42	7200.0
TDS_R107	395.92	410.42	40.7	401.08	401.08	830.4	604.52	612.42	6325.6	604.52	612.42	7200.0
TDS_R108	348.86	357.92	142.3	350.42	350.42	575.6		539.83	7200.0			
TDS_R109	386.35	389.83	57.8	389.83	389.83	321.3	627.58	646.67	2171.7	627.58	646.67	7200.0
TDS_R110	354.31	354.42	289.4	354.42	354.42	709.3		641.50	7200.0			
TDS_R111	384.67	390.33	69.9	387.67	387.67	663.3		640.25	7200.0			
TDS_R112	332.13	349.75	343.1	337.33	337.33	3243.1		546.83	7200.0			
TDS_R201	463.58	463.58	47.7	463.58	463.58	48.1	800.42	805.00	1109.6	800.54	805.00	7200.0
TDS_R202	410.75	410.75	37.8	410.75	410.75	38.2	714.58	726.42	2176.4	714.58	726.42	7200.0
TDS_R203	399.93	400.58	44.7	400.58	400.58	126.8	619.06	635.42	6493.1	619.06	635.42	7200.0
TDS_R204	353.71	359.25	232.9	358.83	358.83	2351.1		546.75	7200.0			
TDS_R205	395.17	395.17	56.9	395.17	395.17	57.4	691.38	695.75	2619.6	691.38	695.75	7200.0
TDS_R206	376.33	388.75	118.8	378.08	378.08	911.3		736.58	7200.0			
TDS_R207	367.17	367.17	127.9	367.17	367.17	128.3		605.75	7200.0			
TDS_R208	339.29	341.08	312.9	341.08	341.08	1763.6		545.75	7200.0			
TDS_R209	385.49	387.83	108.9	387.83	387.83	1113.0	611.29	617.58	6296.9	611.29	617.58	7200.0
TDS_R210	408.56	414.58	89.4	411.75	411.75	703.4		663.67	7200.0			
TDS_R211	351.17	351.17	80.0	351.17	351.17	80.7		640.50	7200.0			
TDS_RC101	358.25	358.25	13.4	358.25	358.25	13.8	632.58	632.58	259.8	632.58	632.58	262.9
TDS_RC102	335.92	335.92	53.8	335.92	335.92	54.3	602.25	602.25	1998.6	602.25	602.25	2002.5
TDS_RC103	327.08	327.08	134.4	327.08	327.08	134.9	584.67	584.67	4736.1	584.67	584.67	4740.2
TDS_RC104	299.75	299.75	709.5	299.75	299.75	709.9		522.92	7200.0			
TDS_RC105	334.75	334.75	35.0	334.75	334.75	35.6	616.17	616.17	1246.5	616.17	616.17	1251.4
TDS_RC106	310.83	310.83	84.9	310.83	310.83	85.4	564.92	564.92	1137.9	564.92	564.92	1143.0
TDS_RC107	296.33	296.33	644.4	296.33	296.33	645.2	522.67	522.67	5132.1	522.67	522.67	5138.8
TDS_RC108	294.50	294.50	5848.2	294.50	294.50	5849.1		517.67	7200.0			
TDS_RC201	360.50	360.50	24.5	360.50	360.50	24.8	685.17	685.17	551.4	685.17	685.17	554.1
TDS_RC202	338.17	338.17	79.8	338.17	338.17	80.2	617.17	617.17	627.9	617.17	617.17	631.2
TDS_RC203	327.08	327.08	33.0	327.08	327.08	33.5	598.83	598.83	1383.3	598.83	598.83	1386.7
TDS_RC204	299.75	299.75	88.2	299.75	299.75	88.7		496.00	7200.0			
TDS_RC205	338.08	338.08	29.5	338.08	338.08	30.0	630.33	630.33	535.9	630.33	630.33	539.0
TDS_RC206	324.25	324.25	70.9	324.25	324.25	71.5	610.17	610.17	681.0	610.17	610.17	684.6
TDS_RC207	298.33	298.33	48.0	298.33	298.33	48.6	560.00	560.00	1856.8	560.00	560.00	1861.2
TDS_RC208	289.75	294.50	578.3	290.96	294.50	7200.0		518.50	7200.0			
Average			610.4						4679.6			

Table 8 Results for EU regulations

	Acceleration factor	
	25 customers	50 customers
US (before 2013)	1.1	1.1
US (after 2013)	1.5	1.5
EU	5.0	6.3

Table 9 Acceleration obtained with heuristic dominance

parameter-free and allow the elimination of many partial paths due to dominance rules exploiting the property that all REFs are non-decreasing. Our approach successfully solves all 25 customer instances for old and new hours of service regulations in the United States and 53 of 56 instances for EU regulations. Furthermore, several of the 50 and 100 customer instances are solved to optimality, however, many others remain open. For the future, we hope that the presented findings are helpful for the development of exact as well as heuristic approaches for VRTDSP with additional constraints or different regulations.

Appendix

This appendix describes resource values, REFs, and a parameter-free network that can be used within the solution approach presented in this paper to solve the VRTDSP for all EU rules described in Section 4.5. The main parameters of these rules are given in Table 10.

Symbol	Value	Description
$t^{\text{drive}\|\text{B}}$	$4\frac{1}{2}$ hours	The maximum driving time without a break or rest
$t^{\text{drive}\|\text{R}}$	9 hours	The maximum driving time without a rest
t^{break}	$\frac{3}{4}$ hours	The minimum duration of a break
$t^{\text{break}\|\text{1st}}$	$\frac{1}{4}$ hour	The minimum duration of the first part of a break taken in two parts
$t^{\text{break}\|\text{2nd}}$	$\frac{1}{2}$ hour	The minimum duration of the second part of a break taken in two parts
t^{rest}	11 hours	The minimum duration of a rest period
$t^{\text{rest}\|\text{1st}}$	3 hours	The minimum duration of the first part of a daily rest period taken in two parts
$t^{\text{rest}\|\text{2nd}}$	9 hours	The minimum duration of the second part of a daily rest period taken in two parts
t^{day}	24 hours	The duration of a day

Table 10 Parameters imposed by EU regulations

To solve the VRTDSP subject to these regulations, we can represent a resource tuple as

$$l = \left(l^{\text{cost}}, l^{\text{load}}, l^{\text{time}}, l^{\text{dist}}, l^{\text{drive}|\text{R}}, l^{\text{drive}|\text{B}}, l^{\text{rest}}, l^{\text{break}}, l^{\text{elapsed}|\text{R}}, l^{\text{latest}|\text{R}}\right),$$

where the resource values l^{cost}, l^{load}, l^{time}, l^{dist}, $l^{\text{drive}|\text{R}}$, $l^{\text{elapsed}|\text{R}}$, and $l^{\text{latest}|\text{R}}$ have the same interpretation as for U.S. regulations. The semantics of the other resources is as follows:

$l^{\text{drive|B}}$: the total amount of driving since the last break or rest;

l^{rest}: the minimum duration of the next rest, i.e., $t^{\text{rest|2nd}}$ if the first part of a rest is already taken or t^{rest} otherwise;

l^{break}: the minimum duration of the next break, i.e., $t^{\text{break|2nd}}$ if the first part of a break is already taken or t^{break} otherwise.

These new resource values are updated by the REFs as shown in Table 11. The two new REFs $f_\Delta^{\text{break|1st}}$ and $f_\Delta^{\text{rest|1st}}$ are needed to consider the possibility of taking the first part of a break or the first part of a rest and to change the minimum duration required to complete the break or rest.

	Resource extension functions (REFs)							
Result	$f_{nm}^{\text{start}}(l)$	$f_\Delta^{\text{drive}}(l)$	$f_\Delta^{\text{wait}}(l)$	$f_\Delta^{\text{break\|1st}}(l)$	$f_\Delta^{\text{break}}(l)$	$f_\Delta^{\text{rest\|1st}}(l)$	$f_\Delta^{\text{rest}}(l)$	$f_{nm}^{\text{visit}}(l)$
$\hat{l}^{\text{drive\|B}} :=$		$l^{\text{drive\|B}} + \Delta$			0	0	0	
$\hat{l}^{\text{rest}} :=$						$t^{\text{rest\|2nd}}$	t^{rest}	
$\hat{l}^{\text{break}} :=$				$t^{\text{break\|2nd}}$	t^{break}	t^{break}	t^{break}	

Table 11 Resource extension functions for EU regulations

The remaining resource values are updated by the REFs in the same way as for U.S. regulations where $f_\Delta^{\text{break|1st}}$ is equivalent to f_Δ^{wait} and $f_\Delta^{\text{rest|1st}}$ is equivalent to f_Δ^{break} for those resource values. For all REFs f we have to add the feasibility condition $\big(f(l)\big)^{\text{elapsed|R}} + \big(f(l)\big)^{\text{rest}} \le t^{\text{day}}$ because otherwise it would not be possible to take the next rest within 24 hours after the end of the previous rest. Thus, the maximum value for Δ when using REF $f_\Delta^{\text{drive}}(l)$ is

$$\Delta_l := \min\left\{l^{\text{dist}}, t^{\text{drive|R}} - l^{\text{drive|R}}, t^{\text{drive|B}} - l^{\text{drive|B}}, t^{\text{day}} - (l^{\text{elapsed|R}} + l^{\text{rest}})\right\}.$$

The minimum values for Δ when using REFs $f_\Delta^{\text{break|1st}}(l)$, $f^{\text{break}}(l)$, $f_\Delta^{\text{rest|1st}}(l)$, and $f_\Delta^{\text{rest}}(l)$ are $t^{\text{break|1st}}$, l^{break}, $t^{\text{rest|1st}}$, and l^{rest}.

With these resource values and REFs we can formulate the parameter-free network shown in Figure 6. This network can be derived with similar reasoning as for U.S. regulations and previous work on European regulations (Goel 2010, Drexl and Prescott-Gagnon 2010, Prescott-Gagnon et al. 2010). For any arc without text in the figure, no resource value is changed when using the arc. The main difference to the network for U.S. regulations is that before arriving at the next customer, the first part of a rest can be taken instead of a break, and that after arriving at the next customer, any reasonable combination of taking breaks and rest periods is considered before applying REF f_{nm}^{visit}. It must be noted that, when calculating the lower bound on the duration of a trip along an arc $(n,m) \in A$, the durations of the first break and rest have to be reduced to $t^{\text{break|2nd}}$ and $t^{\text{rest|2nd}}$ because the first part of the break or rest may have been taken before starting the trip.

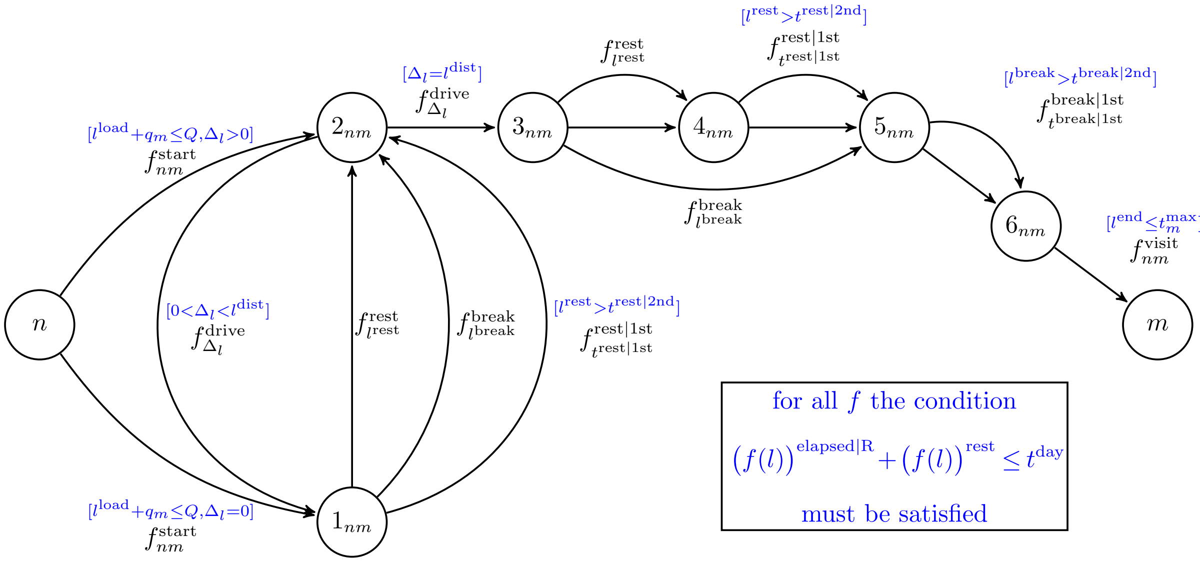

Figure 6 **Parameter-free model for EU regulations**

References

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

Baldacci, R., A. Mingozzi, R. Roberti. 2011b. New state-space relaxations for solving the traveling salesman problem with time windows. *INFORMS Journal on Computing* **24**(3) 356–371. doi:10.1287/ijoc.1110.0456.

Baldacci, R., A. Mingozzi, R. Roberti. 2012. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research* **218**(1) 1–6. doi:10.1016/j.ejor.2011.07.037.

Bartodziej, P., U. Derigs, D. Malcherek, U. Vogel. 2009. Models and algorithms for solving combined vehicle and crew scheduling problems with rest constraints: An application to road feeder service planning in air cargo transportation. *OR Spectrum* **31** 405–429.

Beasley, J.E., N. Christofides. 1989. An algorithm for the resource constrained shortest path problem. *Networks* **19** 379–394.

Bode, C., S. Irnich. 2014. In-depth analysis of pricing problem relaxations for the capacitated arc-routing problem. *Transportation Science* **49**(2). doi:10.1287/trsc.2013.0507.

Boland, N., J. Dethridge, I. Dumitrescu. 2006. Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Operations Research Letters* **34**(1) 58–68.

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.

Clarke, G., J. W. Wright. 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* **12**(4) 568–581.

Cordeau, J.-F., G. Desaulniers, J. Desrosiers, M. Solomon, F. Soumis. 2002. VRP with time windows. P. Toth, D. Vigo, eds., *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 157–193.

Derigs, U., R. Kurowsky, U. Vogel. 2011. Solving a real-world vehicle routing problem with multiple use of tractors and trailers and EU-regulations for drivers arising in air cargo road feeder services. *European Journal of Operational Research* **213** 309–319.

Desaulniers, G., J. Desrosiers, I. Ioachim, M.M. Solomon, F. Soumis, D. Villeneuve. 1998. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. T.G. Crainic, G. Laporte, eds., *Fleet Management and Logistics*, chap. 3. Kluwer Academic Publisher, Boston, Dordrecht, London, 57–93.

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

Desaulniers, G., J. Desrosiers, S. Spoorendonk. 2010. *The Vehicle Routing Problem with Time Windows: State-of-the-Art Exact Solution Methods*. John Wiley & Sons, Inc.

Desaulniers, G., F. Lessard, A. Hadjar. 2008. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science* **42**(3) 387–404.

Desaulniers, G., O.B.G. Madsen, S. Ropke. 2014. The vehicle routing problem with time windows. P. Toth, D. Vigo, eds., *Vehicle Routing, Problems, Methods, and Applications*. MOS-SIAM Series on Optimization, SIAM Society for Industrial and Applied Mathematics, Philadelphia, PA, 119–159.

Desrochers, M. 1986. La fabrication d'horaires de travail pour les conducteurs d'autobus par une méthode de génération de colonnes. Ph.D. thesis, Centre de recherche sur les Transports, Université de Montréal, Montréal, Canada. Publication #470, in French.

Desrochers, M., J. Desrosiers, M. Solomon. 1992. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research* **40**(2) 342–354.

Drexl, M., E. Prescott-Gagnon. 2010. Labelling Algorithms for the Elementary Shortest Path Problem with Resource Constraints Considering EU Drivers' Rules. *Logistics Research* **2** 79–96.

Dror, Moshe. 1994. Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research* **42**(5) 977.

Federal Motor Carrier Safety Administration. 2008. Hours of service of drivers. Federal Register Vol. 73, No. 224, p. 69569 - 69586.

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

Feillet, D. 2010. A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR* **8**(4) 407–424. doi:10.1007/s10288-010-0130-z.

Feillet, D., P. Dejax, M. Gendreau, C. Gueguen. 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks* **44**(3) 216–229. doi:10.1002/net.20033.

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.

Goel, A. 2010. Truck driver scheduling in the European Union. *Transportation Science* **44**(4) 429–441.

Goel, A. 2012. The minimum duration truck driver scheduling problem. *EURO Journal on Transportation and Logistics* **1**(4) 285–306. 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., C. Archetti, M. Savelsbergh. 2012. Truck driver scheduling in Australia. *Computers & Operations Research* **39**(5) 1122–1132.

Goel, A., L. Kok. 2012a. Efficient scheduling of team truck drivers in the European Union. *Flexible Services and Manufacturing Journal* **24**(1) 81–96.

Goel, A., L. Kok. 2012b. Truck driver scheduling in the United States. *Transportation Science* **46**(3) 317–326.

Goel, A., L. M. Rousseau. 2012. Truck driver scheduling in Canada. *Journal of Scheduling* **15**(6) 783–799.

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

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. G. Desaulniers, J. Desrosiers, M.M. Solomon, eds., *Column Generation*, chap. 2. Springer, 33–65.

Irnich, S., D. Villeneuve. 2006. The shortest-path problem with resource constraints and k-cycle elimination for $k \geq 3$. *INFORMS Journal on Computing* **18**(3) 391 – 406.

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.

Lübbecke, M.E., J. Desrosiers. 2005. Selected topics in column generation. *Operations Research* **53**(6) 1007–1023.

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.

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.

Righini, G., M. Salani. 2008. New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks* **51**(3) 155–170. doi:10.1002/net.20212.

Roberti, R., A. Mingozzi. 2014. Dynamic ng-path relaxation for the delivery man problem. *Transportation Science* **48**(3) 413–424. doi:10.1287/trsc.2013.0474.

Savelsbergh, M. W. P., M. Sol. 1998. DRIVE: dynamic routing of independent vehicles. *Operations Research* **46** 474–490.

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

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.