

The minimum duration truck driver scheduling problem

Asvin Goel

Received: 13 January 2012 / Accepted: 27 July 2012 / Published online: 31 August 2012
© Springer-Verlag + EURO - The Association of European Operational Research Societies 2012

Abstract Truck driver scheduling problems are important subproblems of real-life vehicle routing and scheduling problems, because rest periods required by government regulations have a significant impact on travel and arrival times. Vehicle routes generated without considering these regulations are often practically infeasible. This paper identifies common constraints imposed by hours of service regulations world wide and presents a mixed integer programming formulation for a variant of the truck driver scheduling problem in which truck drivers may only rest at customer locations and at suitable rest areas. The objective of the problem is to find a feasible truck driver schedule with minimal duration. The model presented in this paper is very flexible and can be configured to consider different sets of rules imposed by government regulations and union contracts. A dynamic programming approach is presented and its effectiveness is demonstrated for working hour regulations in the United States and in the European Union.

Keywords US hours of service regulations · Regulation (EC) No 561/2006 · Multiple time windows · Rest areas · Vehicle scheduling · Dynamic programming

Introduction

In their efforts to increase road safety and improve working conditions of truck drivers, governments world wide have implemented regulations constraining driving and working hours of truck drivers. Compulsory breaks and rest periods mandated by hours of service regulations have a significant impact on total travel

A. Goel
University of Duisburg-Essen, Essen, Germany

A. Goel (✉)
Zaragoza Logistics Center, Saragossa, Spain
e-mail: agoel@zlc.edu.es; goel@telematique.eu

times which are typically more than twice as long as the pure driving time required in long-distance haulage. Ignoring compulsory breaks and rest periods when generating schedules for truck drivers can lead to unrealistic expectations, large delays, and violations of driving and working hour regulations. This results in poor working conditions of truck drivers, reduced road safety, and low customer satisfaction.

Looking at different hours of service regulations world wide, it can be observed that these regulations typically impose limits on the amount of driving and working between rest periods and the time that may elapse between rest periods of the same type (Federal Motor Carrier Safety Administration 2011; European Union 2006; Transport Canada 2005; National Transport Commission 2008a, b). In some regulations, however, additional differently structured rules must be complied with. In this paper the most commonly found constraints imposed by hours of service regulations are modelled, and a generic mixed integer programming formulation for the problem of finding a truck driver schedule with minimal duration which complies with these constraints is presented. The model can be configured to consider different sets of constraints imposed by different hours of service regulations. In many practical applications, in particular when motorways are used, rest periods cannot be taken anywhere. The model presented in this paper explicitly considers the location of rest areas. Therefore, truck driver schedules satisfying the constraints of the model will be more realistic than schedules generated by previous approaches assuming that rest periods can be taken anywhere.

A dynamic programming approach is presented for solving the problem described above. The dynamic programming approach is very fast and its effectiveness is demonstrated for working hour regulations in the United States and in the European Union.

The remainder of this paper is organised as follows: the next section summarises hours of service regulations in the United States and the European Union. In “[Related work](#)” an overview over the state-of-the-art in truck driver scheduling is given. In “[Problem formulation](#)”, the problem studied in this paper is formulated as a mixed integer program. In “[Solution approach](#)”, the dynamic programming approach is presented and in “[Extensions](#)” it is shown how additional rules can be considered. In “[Computational experiments](#)” experimental results are presented demonstrating the importance of considering the location of rest areas and the effectiveness of the dynamic programming approach. Eventually, conclusions are given in “[Conclusion](#)”.

Hours of service regulations

United States

Present hours of service regulations imposed by the US Department of Transportation are comprehensively described by the Federal Motor Carrier Safety Administration (2011). The regulation distinguishes between *on-duty time* and *off-duty time*. On-duty time refers to all time a driver is working and includes

driving activities as well as other work such as loading and unloading. Off-duty time refers to any time during which a driver is not performing any work.

The regulation limits the maximum amount of accumulated driving time between two rest periods to 11 h. After accumulating 11 h of driving, the driver must take a rest period of 10 consecutive hours before driving again. The regulation prohibits a driver from driving after 14 h have elapsed since the end of the last rest period. However, a driver may conduct other work after 14 h have elapsed since the end of the last rest period. The regulation also prohibits a driver from driving after accumulating 70 h of on-duty time within a week.

In July 2013, an additional rule will become effective which demands that a driver does not drive after 8 h have elapsed since the end of the last off-duty period of at least 30 min.

European Union

In the European Union, working hours of truck drivers are constrained by regulation (EC) No 561/2006 which distinguishes between four driver activities: rest periods, breaks, driving time, and other work. Rest periods are longer periods during which a driver may freely dispose of her or his time. Breaks are short periods exclusively used for recuperation during which a driver may not carry out any work. Driving time refers to the time during which a driver is operating a vehicle and includes any time during which the vehicle is temporarily stationary due to reasons related to driving, e.g. traffic jams. Other work refers to any work except for driving and includes time spent for loading or unloading, cleaning and technical maintenance, customs, etc.

The regulation demands that a driver takes a break of at least 45 min after accumulating four and a half hours of driving. A rest period of at least 11 h must be completed within each period of 24 h after the end of the previous rest period. The accumulated driving time between two rest periods must not exceed 9 h.

The regulation limits the amount of driving within a week to 56 h and the accumulated amount of driving and working in a week to at most 60 h. Furthermore, a driver must not drive or work after 144 h have elapsed since the begin of the work week.

The basic set of rules described above are sufficient to comply with Regulation (EC) No 561/2006. There are several additional rules in Regulation (EC) No 561/2006 which are comprehensively discussed in Goel (2010). These additional rules give drivers more flexibility in scheduling breaks and rest periods. Among them are the rules that breaks and rest periods may be taken in two parts. The first part of a break period taken in two parts must have a duration of at least 15 min, and the second must have a duration of at least 30 min. The first part of a rest period taken in two parts must have a duration of at least 3 h, and the second must have a duration of at least 9 h. If a rest period is taken in two parts, the second part must be completed within 24 h after the end of the previous rest period.

It must be noted that further rules are implemented into national law by all member states of the European Union. These national laws include constraints outlined by Directive 2002/15/EC on the maximum amount of work without a break and on the amount of night work. The directive is not a law itself and the national laws have been differently implemented throughout Europe. A full coverage of all

rules is beyond the scope of this paper and is omitted here for the sake of conciseness.

Related work

Hours of service rules imposed by the US Department of Transportation are first studied by Xu et al. (2003) who present a column generation approach for combined vehicle routing and scheduling. They conjecture that determining a minimal cost truck driver schedule for a given sequence of customer locations is NP-hard in the presence of multiple time windows. Archetti and Savelsbergh (2009) studied a similar problem with single time windows and showed that a feasible truck driver schedule complying with US hours of service regulations for a sequence of n locations to be visited can be found in $O(n^3)$ time. Goel and Kok (2011) show that feasible schedules complying with US hours of service regulations can be found in $O(n^2)$ time. Based on this approach, Rancourt et al. (2010) present a tabu search heuristic for combined vehicle routing and scheduling in the United States.

In December 2011, the Federal Motor Carrier Safety Administrations published new hours of service regulations in the United States. The new regulations introduce additional break constraints which were not included in the previous regulations. These additional break constraints will become effective in July 2013 and are discussed in Goel (2012c).

Working hours of truck drivers in the European Union are controlled by Regulation (EC) No 561/2006. A first comprehensive model of the regulation is given by Goel (2010). An exact method is presented which is guaranteed to find a truck driver schedule complying with the constraints of the model if such a schedule exists. Following this work, Drexl and Prescott-Gagnon (2010) present a scheduling approach using the notion of resource extension functions. Goel (2009), Kok et al. (2010b), and Prescott-Gagnon et al. (2010) present heuristic approaches for solving combined vehicle routing and truck driver scheduling problems in the European Union. European Union regulations are also considered within the works on rich vehicle routing problems presented by Zäpfel and Bögl (2008), Bartodziej et al. (2009), Kok et al. (2010a) and Derigs et al. (2011).

Meyer (2011) and Kok et al. (2011) present mixed integer programming formulations for minimising the duration of truck driver schedules in the European Union. However, the model presented by Meyer (2011) is flawed as the constraints of European Union regulations, which require that a rest period is completed within 24 h after the end of the previous rest period, are modelled in a way demanding that at least k rest periods must be scheduled within k days since the start of the week. If a rest period is scheduled earlier than required, this way of modelling does not guarantee that the next rest period is completed within 24 h after the end of the rest period. Therefore, some schedules satisfying the constraints of the model may violate the regulation. The model presented by Kok et al. (2011) focusses on a planning horizon of at most 13 h (i.e. a working day). An extension for planning horizons of multiple days is outlined, but the possibility of taking rest periods in two

parts is not considered and no computational experiments are presented for planning horizons of several days. Although the model allows to take breaks in two parts, the possibility of taking the first part of a break immediately after completing a break or rest period is not considered. This possibility, however, may be necessary to find a feasible schedule (Goel 2010).

Hours of service regulations in Canada and Australia comprise constraints similar to those in the United States and the European Union; however, they also contain additional constraints having a unique and different structure. The only works known to the author considering these regulations are presented by Goel and Rousseau (2011) and Goel et al. (2012).

All of the above-mentioned approaches are developed for a specific set of rules and each rule is individually modelled. As no effort has yet been made to identify common characteristics of rules, a new method is developed for each set of rules considered.

Except for the approach by Kok et al. (2011) all existing approaches for truck driver scheduling assume that truck drivers may take rest periods anywhere. In this paper, we assume like Kok et al. (2011) that truck drivers may only take rest periods at customer locations and at suitable rest areas. The explicit consideration of locations suitable for a driver to take rest periods is particularly important where there is a shortage of safe and secured rest areas. The European Economic and Social Committee states that such a shortage is prevalent in the European Union and demands that enough suitable parking areas must be made available along European highways (see European Union 2007). If the availability of suitable rest areas is ignored when generating truck driver schedules, a driver required to take a rest period may not be able to find a suitable rest area and must continue driving until such a rest area is found. This can result in excessive driving times and reduced road safety.

Problem formulation

This section presents a generic model for a variant of the truck driver scheduling problem which can be configured to consider different hours of service regulations. In the problem variant studied in this paper the location of suitable rest areas is explicitly considered and the goal is to find a truck driver schedule with minimal duration. To provide a formal model of this problem, let us consider a sequence of n locations which shall be visited by a truck driver. At each location $1 \leq i \leq n$ some stationary work of duration w_i shall be conducted. This work shall begin within one of multiple disjunct time windows. The number of time windows at location $1 \leq i \leq n$ shall be denoted by T_i . For each $1 \leq \tau \leq T_i$ the τ th time window at location i shall be denoted by the interval $[t_{i,\tau}^{\min}, t_{i,\tau}^{\max}]$. The driving time required for moving from location i to $i + 1$ shall be denoted by $d_{i,i+1}$. The time horizon shall be denoted by t^{horizon} . The minimum duration truck driver scheduling problem (MD-TDSP) is the problem of determining a schedule complying with applicable legislation in which all work activities begin within one of the corresponding time windows and

which has the minimal duration, i.e. the time between the beginning of the first work and the end of the last work. As the sequence of locations to be visited as well as the duration of work and the driving time between locations are fixed, the main decisions to be made when solving the MD-TDSP are those regarding the type and duration of rest periods at the different locations.

In this paper, we assume that drivers may only take rest periods after arrival at a location and before starting the work activity at the location. We will later see that the model presented in this section can also be used if drivers may take rest periods after completing the work or at suitable rest areas on the trip from one location to another.

To develop a generic model that can be used for different hours of service regulations, let R denote the set of different types of rest periods defined by the regulation considered. For each $r \in R$, let t^r denote the minimum duration imposed by the regulation. Let C denote a set of constraints imposed by the regulation and let C_1, C_2, C_3 , and C_4 denote four subsets of C which will be defined below. For each constraint $c \in C$, let t^c denote a limit imposed by the regulation and let $R^c \subseteq R$ denote the set of rest types a driver can take to reset the counter bounded by the limit t^c .

Let C_1 denote a set of constraints c which require that a driver may not continue driving after a certain amount of time has elapsed since the end of the last rest period of type $r \in R^c$. Let C_2 denote a set of constraints c which require that a driver may not continue to work after a certain amount of time has elapsed since the end of the last rest period of type $r \in R^c$. Let $C_3, C_4 \subseteq C$ denote sets of constraints c which limit the accumulated amount of driving or the accumulated amount of driving and working in between rest periods of some rest of type $r \in R^c$. For each constraint $c \in C_3 \cup C_4$, let δ^c denote a binary parameter indicating whether working periods in which the driver is not driving are counted ($\delta^c = 1$) for the constraint c or not ($\delta^c = 0$).

In the remainder of this paper we will assume that

$$R^{c'} \subseteq R^{c''} \quad \text{or} \quad R^{c''} \subseteq R^{c'} \quad \text{for any } c', c'' \in C_1 \cup C_2. \tag{A}$$

That is, the constraints in $C_1 \cup C_2$ can be ordered in a way that all rest periods resetting the time elapsed with respect to the highest ordered constraints also reset the time elapsed with respect to the lower ordered constraints. This assumption is not restrictive if constraints are similarly structured to the constraints imposed by the regulations in the United States or the European Union regulations. In these regulations a rest period can also be used to reset the constraints associated with breaks.

We can now provide a mixed integer programming formulation of the MD-TDSP. For each location $1 \leq i \leq n$ the formulation comprises variables $x_i := (x_i^{\text{arrival}}, x_i^{\text{start}}, x_i^{\text{end}})$ representing the arrival time, the start time, and the end time of the work at location i . Furthermore, the formulation comprises for each location $1 \leq i \leq n$ the variables $y_i = (y_{i,\tau})_{1 \leq \tau \leq T_i}$ where $y_{i,\tau}$ is a binary variable indicating whether the τ th time window of location i is used ($y_{i,\tau} = 1$) or not ($y_{i,\tau} = 0$). Eventually, the formulation comprises for each location $1 \leq i \leq n$ the variables

$z_i = (z_{i,r})_{r \in R}$ where $z_{i,r}$ is a binary variable indicating whether a rest of type r is taken at location i ($z_{i,r} = 1$) or not ($z_{i,r} = 0$). The MD-TDSP is minimise

$$x_n^{\text{end}} - x_1^{\text{start}} \tag{1}$$

subject to

$$x_i^{\text{arrival}} + \sum_{r \in R} t^r z_{i,r} \leq x_i^{\text{start}} \quad \text{for all } 1 \leq i \leq n \tag{2}$$

$$x_i^{\text{start}} + w_i = x_i^{\text{end}} \quad \text{for all } 1 \leq i \leq n \tag{3}$$

$$x_i^{\text{end}} + d_{i,i+1} = x_{i+1}^{\text{arrival}} \quad \text{for all } 1 \leq i < n \tag{4}$$

$$\sum_{\tau=1}^{T_i} y_{i,\tau} = 1 \quad \text{for all } 1 \leq i \leq n \tag{5}$$

$$y_{i,\tau} t_{i,\tau}^{\text{min}} \leq x_i^{\text{start}} \quad \text{for all } 1 \leq i \leq n, 1 \leq \tau \leq T_i \tag{6}$$

$$x_i^{\text{start}} \leq t^{\text{horizon}} - y_{i,\tau} (t^{\text{horizon}} - t_{i,\tau}^{\text{max}}) \quad \text{for all } 1 \leq i \leq n, 1 \leq \tau \leq T_i \tag{7}$$

$$x_k^{\text{arrival}} - x_i^{\text{start}} \leq t^c + t^{\text{horizon}} \sum_{j=i+1}^{k-1} \sum_{r \in R^c} z_{j,r} \quad \text{for all } 1 \leq i < k \leq n : d_{k-1,k} > 0, c \in C_1 \tag{8}$$

$$x_k^{\text{end}} - x_i^{\text{start}} \leq t^c + t^{\text{horizon}} \sum_{j=i+1}^k \sum_{r \in R^c} z_{j,r} \quad \text{for all } 1 \leq i < k \leq n : w_k > 0, c \in C_2 \tag{9}$$

$$\sum_{j=i}^{k-1} d_{j,j+1} + \delta^c \sum_{j=i}^{k-1} w_j \leq t^c + t^{\text{horizon}} \sum_{j=i+1}^{k-1} \sum_{r \in R^c} z_{j,r} \quad \text{for all } 1 \leq i < k \leq n : d_{k-1,k} > 0, c \in C_3 \tag{10}$$

$$\sum_{j=i}^{k-1} d_{j,j+1} + \delta^c \sum_{j=i}^k w_j \leq t^c + t^{\text{horizon}} \sum_{j=i+1}^k \sum_{r \in R^c} z_{j,r} \quad \text{for all } 1 \leq i < k \leq n : w_k > 0, c \in C_4 \tag{11}$$

$$\sum_{r \in R} z_{i,r} \leq 1 \quad \text{for all } 1 \leq i \leq n \tag{12}$$

$$x_i \in [0, t^{\text{horizon}}]^3, y_i \in \{0, 1\}^{T_i}, z_i \in \{0, 1\}^{|R|} \quad \text{for all } 1 \leq i \leq n \tag{13}$$

The objective function (1) is to minimise the duration between the start of the first work and the end of the last work. Constraint (2) demands that the work at any location must not start before the arrival plus any rest time. Constraint (3) demands that the work at any location i ends w_i time units after it starts. Constraint (4) demands that the arrival at a location equals the end time of the previous location plus the required driving time. Constraint (5) demands that at any location exactly

one of the time windows is used. Constraints (6) and (7) are the time windows constraints. Constraints (8) and (9) demand that the time elapsed since the end of the last rest in R^c is within the limits imposed by the constraints $c \in C_1$ and $c \in C_2$. Constraints (10) and (11) demand that the accumulated amount of driving and working without rest does not exceed the respective limit imposed by the constraints $c \in C_3$ and $c \in C_4$. Note that constraints (8) to (11) can only be satisfied if $d_{j,j+1} \leq t^c$ for all $1 \leq i < n$ and $c \in C$, i.e., the driving time between two subsequent locations must be small enough such that the limits imposed by the regulation can be satisfied. Constraint (12) demands that at each location at most one rest period is scheduled. If we want to allow more than one rest period at a location we can represent any combination of different rest types by an additional rest type which is introduced to R . The variable domains are given by (13).

Let us now illustrate the constraints of the model for a planning horizon of one week and hours of service regulations in the United States and the European Union.

Current regulations in the United States do not distinguish between different types of rest periods. We have $R = \{\text{rest}\}$ and $t^{\text{rest}} = 10$. The regulation prohibits a driver from driving after 14 h have elapsed since the end of the last rest period. Thus, we have $C_1 = \{\text{elapsed}\}$, $t^{\text{elapsed}} = 14$, and $R^{\text{elapsed}} = \{\text{rest}\}$. The regulation limits the maximum amount of accumulated driving time between two rest periods to 11 h. Furthermore, the regulation also prohibits a driver from driving after accumulating 70 h of on-duty time within a week. Thus, we have $C_3 = \{\text{weekly, daily}\}$, $t^{\text{weekly}} = 70$ and $t^{\text{daily}} = 11$, $\delta^{\text{weekly}} = 1$ and $\delta^{\text{daily}} = 0$, and $R^{\text{weekly}} = \emptyset$ and $R^{\text{daily}} = \{\text{rest}\}$. Furthermore, we have $C_2 = C_4 = \emptyset$. When the new rules become effective in July 2013 we will have to distinguish between breaks and rest periods. Therefore, we add the rest type “break” to R and set $t^{\text{break}} = \frac{1}{2}$. Furthermore, we include the constraint “elapsed_break” in C_1 and set $t^{\text{elapsed_break}} = 8$, and $R^{\text{elapsed_break}} = \{\text{rest, break}\}$. Nothing else needs to be changed.

The basic set of rules in the European Union also distinguishes between rest periods and breaks. We have $R = \{\text{rest, break}\}$, $t^{\text{rest}} = 11$, and $t^{\text{break}} = \frac{3}{4}$. A rest period must be completed within each period of 24 h after the end of the previous rest period and no driving and working may be conducted after 144 h have passed since the begin of the work week. Thus, we have $C_1 = C_2 = \{\text{elapsed, week}\}$, $t^{\text{elapsed}} = 24 - t^{\text{rest}} = 13$, $t^{\text{week}} = 144$, and $R^{\text{elapsed}} = \{\text{rest}\}$ and $R^{\text{week}} = \emptyset$. The regulation demands that a driver takes a break or rest period after accumulating four and a half hours of driving and that at most 9 h of driving are conducted between two rest periods. Thus, we have $C_3 = \{\text{daily, nonstop}\}$, $C_4 = \emptyset$, $t^{\text{daily}} = 9$, $t^{\text{nonstop}} = 4\frac{1}{2}$, $\delta^{\text{daily}} = \delta^{\text{nonstop}} = 0$, and $R^{\text{daily}} = \{\text{rest}\}$ and $R^{\text{nonstop}} = \{\text{rest, break}\}$.

Similar to Prescott-Gagnon et al. (2010), we can configure the model in such a way that the constraints of Directive 2002/15/EC concerning the maximum amount of work without a break is complied with. In the approach presented by Prescott-Gagnon et al. (2010), a driver must not accumulate more than 6 h of driving and working time without taking a break period. This constraint can be easily modelled by adding the constraint “work” to C_3 and C_4 and by setting $t^{\text{work}} = 6$, $\delta^{\text{work}} = 1$, and $R^{\text{work}} = \{\text{rest, break}\}$. As Directive 2002/15/EC comprises further rules which

are differently implemented throughout Europe we will not consider this extension in the remainder of this work.

Figure 1 illustrates a truck driver schedule which complies with the current US regulations. The largest left-hand side value of condition (8) for which the right-hand side value is t^{elapsed} is achieved for $i = 1$ and $k = 4$. As $x_4^{\text{arrival}} - x_1^{\text{start}} = 14 - 0 = 14 \leq t^{\text{elapsed}} = 14$, condition (8) is satisfied for all $1 \leq i < k \leq n$ with $d_{k-1,k} > 0$, $c \in C_1$. The largest left-hand side value of condition (10) for which the right-hand side value is t^{daily} is also achieved for $i = 1$ and $k = 4$. As $\sum_{j=1}^3 d_{j,j+1} + 0 \cdot \sum_{j=1}^3 w_j = 4 + 3 + 4 + 0 \cdot (1 + 1 + 1) = 11 \leq t^{\text{daily}} = 11$, condition (10) is satisfied for all $1 \leq i < k \leq n$ and $c \in C_3$.

For the new hours of service regulations in the United States which become effective in July 2013, condition (8) is violated, because we have $x_3^{\text{arrival}} - x_1^{\text{start}} = 9 - 0 = 9$ which is greater than $t^{\text{elapsed_break}} = 8$.

The schedule violates European Union rules in several ways. We have $x_4^{\text{arrival}} - x_1^{\text{start}} = 14 - 0 = 14$ which is greater than $t^{\text{elapsed}} = 13$ and thus condition (8) is violated. Furthermore, we have $\sum_{j=1}^2 d_{j,j+1} + 0 \cdot \sum_{j=1}^2 w_j = 4 + 3 + 0 \cdot (1 + 1) = 7$ and $\sum_{j=1}^3 d_{j,j+1} + 0 \cdot \sum_{j=1}^3 w_j = 4 + 3 + 4 + 0 \cdot (1 + 1 + 1) = 11$. As these values exceed $t^{\text{nonstop}} = 4\frac{1}{2}$ and $t^{\text{daily}} = 9$, condition (10) is violated. Last, we have $x_4^{\text{arrival}} + \sum_{r \in R} t^{r,z_{4,r}} = 14 + \frac{3}{4} \cdot 0 + 11 \cdot 1 = 25$ which is larger than $x_4^{\text{start}} = 24$ and thus condition (2) is violated.

The model only allows drivers to take rest periods after arrival at a location and before beginning to work. If the driver may take rest periods after the completion of work activities, we can modify the tour by adding a dummy location with zero working time and an unbounded time window immediately after each customer location. If the driver may take rest periods at suitable rest areas on the trip from one location to another, we can similarly add dummy locations with zero working time and an unbounded time window on the trip. In the latter case the driving time between customer locations and rest areas must be set appropriately.

The MD-TDSP presented in this section can be solved by any mixed integer programming solver. We will, however, see that the MD-TDSP can be solved much faster with the dynamic programming approach presented in the next section. The dynamic programming approach, furthermore, has the advantage that it can be easily adapted to exploit the additional freedom in scheduling rest periods resulting from rules not considered in the formulation presented in this section.

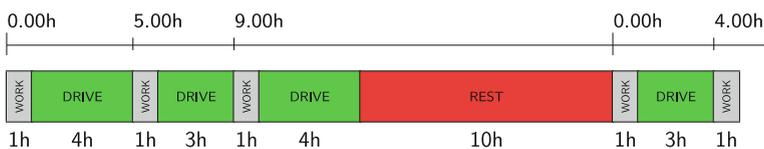


Fig. 1 A truck driver schedule

Solution approach

This section presents a dynamic programming approach (see, e.g., Dasgupta et al. 2007, chap. 6) for solving the MD-TDSP for a tour of length n . The dynamic programming approach starts with a tour of length $j = 1$ and determines a set of feasible solutions $S_j = \{(x_i, y_i, z_i)_{1 \leq i \leq j}\}$ for this tour. Then it uses S_j to find a set of feasible solutions $S_{j+1} = \{(x_i, y_i, z_i)_{1 \leq i \leq j+1}\}$, increments j and repeats this process until $j = n$.

Assume we have a feasible solution $(x_i, y_i, z_i)_{1 \leq i \leq j} \in S_j$. For each combination of values of y_{j+1} and z_{j+1} which satisfies (5) and (12) we can determine new time values x_{j+1} using the function `extend(·)` given in Fig. 2. The function determines the arrival time at location $j + 1$ by adding the driving time to the end time of the previous location. The start time is set to the smallest value which is not before the opening of the time window specified by y_{j+1} and which allows enough time to take the rest specified by z_{j+1} . The end time is the start time plus the duration of the work to be performed at location $j + 1$.

If $x_{j+1}^{\text{start}} > \sum_{\tau=1}^{T_{j+1}} t_{j+1,\tau}^{\max} y_{j+1,\tau}$ the specified time window is violated. Otherwise, $(x_i, y_i, z_i)_{1 \leq i \leq j+1}$ satisfies constraints (2) to (7) of the MD-TDSP. If $(x_i, y_i, z_i)_{1 \leq i \leq j+1}$ violates constraints (10) or (11), the rest pattern defined by $(z_i)_{1 \leq i \leq j+1}$ is infeasible, because the accumulated driving and working time between rest periods exceeds the maximum allowed value. If constraints (2)–(7) and (10)–(13) are satisfied, we only need to check whether constraints (8) and (9) are satisfied or not. If constraints (8) or (9) are violated, then the time that has elapsed between the start of the i th work activity and the arrival at location $j + 1$, or the time that has elapsed between the start of the i th work activity and the completion of work at location $j + 1$ exceeds t^c for some $i < j + 1$ and $c \in C_1 \cup C_2$. The time that has elapsed can be reduced by increasing the start time of the i th work activity. If this can be done without violating any of the other constraints a feasible solution can be obtained.

-
1. $x_{j+1}^{\text{arrival}} = x_j^{\text{end}} + d_{j,j+1}$
 2. $x_{j+1}^{\text{start}} = \max \left\{ x_{j+1}^{\text{arrival}} + \sum_{r \in R} t^r z_{j+1,r}, \sum_{\tau=1}^{T_{j+1}} t_{j+1,\tau}^{\min} y_{j+1,\tau} \right\}$
 3. $x_{j+1}^{\text{end}} = x_{j+1}^{\text{start}} + w_{j+1}$
 4. return $(x_i, y_i, z_i)_{1 \leq i \leq j+1}$
-

Fig. 2 `extend((x_i, y_i, z_i) $_{1 \leq i \leq j}, y_{j+1}, z_{j+1})$`

1. for all $1 \leq i < k$ do

$$\bar{x}_i^{\text{arrival}} = x_i^{\text{arrival}}, \bar{x}_i^{\text{start}} = x_i^{\text{start}}, \bar{x}_i^{\text{end}} = x_i^{\text{end}}$$

2. $\bar{x}_k^{\text{arrival}} = x_k^{\text{arrival}}, \bar{x}_k^{\text{start}} = x_k^{\text{start}} + \Delta, \bar{x}_k^{\text{end}} = \bar{x}_k^{\text{start}} + w_k$

3. for all $k < i \leq j$ do

$$\bar{x}_i^{\text{arrival}} = \bar{x}_{i-1}^{\text{end}} + d_{i-1,i}, \bar{x}_i^{\text{start}} = \max\{x_i^{\text{start}}, \bar{x}_i^{\text{arrival}} + \sum_{r \in R} t^r z_{i,r}\}, \bar{x}_i^{\text{end}} = \bar{x}_i^{\text{start}} + w_i$$

4. return $(\bar{x}_i, y_i, z_i)_{1 \leq i \leq j}$

Fig. 3 $\text{increase}((x_i, y_i, z_i)_{1 \leq i \leq j}, k, \Delta)$

Let us now describe a method that can be used to increase the start time of some work activity to achieve compliance with constraints (8) and (9). The method $\text{increase}(\cdot)$ given in Fig. 3 takes a solution $(x_i, y_i, z_i)_{1 \leq i \leq j}$, an index $k < j$ and a value $\Delta \geq 0$ and determines a new solution in which the start time of the work at location k is increased by Δ . All time values before the start time of the work at location k remain unchanged and all time values, thereafter, are increased by the smallest amount possible.

Let us now introduce some notation that will be used to determine the parameters given to the method $\text{increase}(\cdot)$. For any solution $s = (x_i, y_i, z_i)_{1 \leq i \leq j}$ and any constraint $c \in C_1 \cup C_2$ let us denote with

$$k_s^c = \max\{k \mid \exists r \in R^c : z_{k,r} = 1\}$$

the index of the last location at which a rest period in R^c is taken. The accumulated amount of off-duty time after conducting the stationary work at the location with index k_s^c is $\sum_{i=k_s^c+1}^j (x_i^{\text{start}} - x_i^{\text{arrival}})$. This accumulated off-duty time includes for each location $k_s^c < i \leq j$ the amount of rest as required by $(z_i)_{1 \leq i \leq j}$. The off-duty time exceeding $\sum_{r \in R} t^r z_{i,r}$ at location i is not required by constraint (2). Let us denote this unnecessary off-duty time with

$$l_s^{\text{slack}|c} = \sum_{i=k_s^c+1}^j \left(x_i^{\text{start}} - x_i^{\text{arrival}} - \sum_{r \in R} t^r z_{i,r} \right).$$

We can increase the start time of the work at location k_s^c by any value of not more than $l_s^{\text{slack}|c}$ without violating constraints (8) and (9) and without increasing the end time of the work at location j . Furthermore, let us denote with

$$l_s^{\text{push}|c} = \min_{k_s^c \leq i \leq j} \left\{ \sum_{\tau=1}^{T_i} t_{i,\tau}^{\max} y_{i,\tau} - x_i^{\text{start}} + \sum_{i'=k_s^c+1}^i \left(x_{i'}^{\text{start}} - x_{i'}^{\text{arrival}} - \sum_{r \in R} t^r z_{i',r} \right) \right\}$$

the largest amount by which the start time of the work at location k_s^c can be increased without pushing any subsequent work out of the time window specified by $(y_i)_{1 \leq i \leq j}$. If we increase the start time of the work at location k_s^c by $\Delta = \min\{l_s^{\text{slack}|c}, l_s^{\text{push}|c}\}$, we can reduce t^c by Δ without violating time windows. Note that the end time of the j th work is not increased as $\Delta \leq l_s^{\text{slack}|c}$. Any larger increase would either push one of the start times out of the specified time window or would increase the end time of the work at location j .

The order of constraints $c \in C_1 \cup C_2$ resulting from (A) coincides with the order of k_s^c . If c is the constraint with the smallest k_s^c with $\min\{l_s^{\text{slack}|c}, l_s^{\text{push}|c}\} > 0$, we can increase the start time of the work at location k_s^c by $\Delta = \min\{l_s^{\text{slack}|c}, l_s^{\text{push}|c}\}$ without increasing $t^{c'}$ for any other constraint $c' \in C_1 \cup C_2$. The method `update(·)` given in Fig. 4 takes a solution $s = (x_i, y_i, z_i)_{1 \leq i \leq j}$ and increases the start time of locations until $l_s^{\text{slack}|c} = 0$ or $l_s^{\text{push}|c} = 0$ for all $c \in C_1 \cup C_2$.

If the solution obtained by invoking the method `update(·)` satisfies (8) and (9), a feasible solution is obtained. Otherwise, feasibility cannot be achieved by further increasing start times, because any further increase would either violate time windows or would leave t^c unchanged for any $c \in C_1 \cup C_2$.

Let us now present a solution approach for solving the MD-TDSP with the additional restriction that $(y_i)_{1 \leq i \leq n}$ and $(z_i)_{1 \leq i \leq n}$ are fixed to specific values. Figure 5 outlines a dynamic programming approach for this restricted problem. The approach starts by setting $j = 1$ and determines a solution s_j in which arrival time, start time, and end time have the smallest possible values. Then it determines a solution s_{j+1} using the methods `extend(·)` and `update(·)`. If s_{j+1} is feasible, the method increments j and repeats the previous step. If a feasible solution s_n is found

-
1. $s = (x_i, y_i, z_i)_{1 \leq i \leq j}$
 2. while there is a $c \in C_1 \cup C_2$ with $\min\{l_s^{\text{slack}|c}, l_s^{\text{push}|c}\} > 0$ do
 - (a) select $c \in C_1 \cup C_2$ with $\min\{l_s^{\text{slack}|c}, l_s^{\text{push}|c}\} > 0$ which has the smallest k_s^c
 - (b) $s \leftarrow \text{increase}(s, k_s^c, \min\{l_s^{\text{slack}|c}, l_s^{\text{push}|c}\})$
 3. return s
-

Fig. 4 `update((xi, yi, zi)1 ≤ i ≤ j)`

the method terminates. In any iteration the solution obtained must be checked for feasibility. After invoking the method `extend(·)` constraints (7) to (11) may be violated. If constraint (7) is violated no feasible solution exists because the specified time window is violated although the arrival time at the location is set to the smallest possible value. If constraints (10) or (11) are violated no feasible solution exists, because the accumulated driving and working time between rest periods exceeds the maximum allowed value. If constraints (8) or (9) are violated after invoking the method `update(·)` then no feasible solution exists because t^c exceeds the maximum allowed value for some $c \in C_1 \cup C_2$ and cannot be further reduced. Thus, if no feasible solution is found in any iteration, then no feasible solution exists for given $(y_i)_{1 \leq i \leq n}$ and $(z_i)_{1 \leq i \leq n}$, and the method is prematurely aborted.

The dynamic programming approach for the restricted problem finds a feasible solution for the given values of $(y_i)_{1 \leq i \leq n}$ and $(z_i)_{1 \leq i \leq n}$ if such a solution exists. By definition of the approach all arrival, start, and end times of a solution have the smallest possible values. The duration of a solution obtained by the algorithm, however, may not be minimal. As $(y_i)_{1 \leq i \leq n}$ and $(z_i)_{1 \leq i \leq n}$ are fixed, the start times at all locations are restricted to one of the time windows and the amount of rest to be taken at any location i must be at least $\sum_{r \in R} t^r z_{i,r}$. Unnecessary off-duty times can be removed from a solution by increasing the start time of the work at the first location. For the solution $s = (x_i, y_i, z_i)_{1 \leq i \leq n}$, let

-
1. $j = 1$
 2. $x_1^{\text{arrival}} = 0, x_1^{\text{start}} = \max \left\{ x_1^{\text{arrival}} + \sum_{r \in R} t^r z_{1,r}, \sum_{\tau=1}^{T_1} t_{1,\tau}^{\min} y_{1,\tau} \right\}, x_1^{\text{end}} = x_1^{\text{start}} + w_1$
 3. $s_1 = (x_1, y_1, z_1)$
 4. while $j < n$ do
 - (a) $s_{j+1} = \text{update}(\text{extend}(s_j, y_{j+1}, z_{j+1}))$
 - (b) if s_{j+1} is infeasible then stop - no feasible solution exists for $(y_i)_{1 \leq i \leq n}$ and $(z_i)_{1 \leq i \leq n}$
 - (c) $j \leftarrow j + 1$
 5. return s_n
-

Fig. 5 `restrictedDP((y_i)_{1 \leq i \leq n}, (z_i)_{1 \leq i \leq n})`

$$l_s^{\text{slack}^*} = \sum_{i=2}^n \left(x_i^{\text{start}} - x_i^{\text{arrival}} - \sum_{r \in R} t^r z_{i,r} \right)$$

denote the accumulated amount of off-duty time after start of the first work reduced by the minimum amount of rest required. Furthermore, let

$$l_s^{\text{push}^*} = \min_{1 \leq i \leq j} \left\{ \sum_{\tau \in T_i} t_{i,\tau}^{\text{max}} y_{i,\tau} - x_i^{\text{start}} + \sum_{i'=2}^i \left(x_{i'}^{\text{start}} - x_{i'}^{\text{arrival}} - \sum_{r \in R} t^r z_{i',r} \right) \right\}$$

denote the maximum amount by which we can increase the start of the first work without pushing any subsequent work out of the time window specified by $(y_i)_{1 \leq i \leq n}$.

If s is a feasible solution obtained by the restricted dynamic programming approach, then $s^* = \text{increase}(s, 1, \min\{l_s^{\text{slack}^*}, l_s^{\text{push}^*}\})$ is feasible and has a duration which is $\min\{l_s^{\text{slack}^*}, l_s^{\text{push}^*}\}$ smaller than the duration of s . If $l_s^{\text{push}^*} < l_s^{\text{slack}^*}$, then it is impossible to increase the start time of s^* without violating time window constraints and the start time of s^* is therefore maximal given the values of $(y_i)_{1 \leq i \leq n}$ and $(z_i)_{1 \leq i \leq n}$. As the end time is not increased, s^* has the minimal duration of all feasible solutions for the restricted problem. If $l_s^{\text{push}^*} \geq l_s^{\text{slack}^*}$, we have $l_{s^*}^{\text{slack}^*} = 0$ and the duration of the solution is minimal because no unnecessary off-duty time is scheduled. Thus, for given $(y_i)_{1 \leq i \leq n}$ and $(z_i)_{1 \leq i \leq n}$ the solution s^* is an optimal solution of the restricted problem.

We can now formulate a dynamic programming approach for solving the (unrestricted) MD-TDSP. The approach is given in Fig. 6 and begins with a tour length of $j = 1$. It initialises the sets of partial solutions S_1 by determining a solution with minimal arrival, start, and end times for the first location for each value of y_1 and z_1 satisfying conditions (5) and (12). As long as $j < n$ the approach computes new solutions for the tour of length $j + 1$ in the same way as the restricted dynamic programming approach for each $s_j \in S_j$ and each value of $(y_i)_{1 \leq i \leq j+1}$ and $(z_i)_{1 \leq i \leq j+1}$ satisfying conditions (5) and (12). All new solutions which are feasible are included in the set S_{j+1} . If $j < n$, then j is incremented and the previous steps are repeated. Otherwise, the set S_n is returned. If S_n is empty the problem is infeasible. Otherwise, let s^* denote the solution in $\{\text{increase}(s, 1, \min\{l_s^{\text{slack}^*}, l_s^{\text{push}^*}\}) \mid s \in S_n\}$ which has the minimal duration. As each schedule in this set has the minimal duration for specific values of $(y_i)_{1 \leq i \leq n}$ and $(z_i)_{1 \leq i \leq n}$, and as all feasible values of $(y_i)_{1 \leq i \leq n}$ and $(z_i)_{1 \leq i \leq n}$ are considered, the solution s^* solves the MD-TDSP to optimality.

The approach may generate up to $(|R| + 1)^n \cdot \prod_{1 \leq i \leq n} T_i$ different solutions, i.e. one solution for each combination of $(y_i)_{1 \leq i \leq n}$ and $(z_i)_{1 \leq i \leq n}$. Even if only one type of rest periods must be considered and only one time window is associated with each location (i.e. $|R| = 1, T_i = 1$ for all $1 \leq i \leq n$), the number of solutions generated by the algorithm may theoretically grow exponentially with respect to the tour length n .

1. $S_j = \emptyset$ for all $1 \leq j \leq n, j = 1$
2. for all y_1 and z_1 satisfying (5) and (12) do
 - (a) $x_1^{\text{arrival}} = 0, x_1^{\text{start}} = \max \left\{ x_1^{\text{arrival}} + \sum_{r \in R} t^r z_{1,r}, \sum_{\tau=1}^{T_1} t_{1,\tau}^{\text{min}} y_{1,\tau} \right\},$
 $x_1^{\text{end}} = x_1^{\text{start}} + w_1$
 - (b) $S_1 \leftarrow S_1 \cup \{(x_1, y_1, z_1)\}$
3. while $j < n$ do
 - (a) for all $s_j \in S_j$ and all y_{j+1} and z_{j+1} satisfying (5) and (12) do
 - i. $s_{j+1} = \text{update}(\text{extend}(s_j, y_{j+1}, z_{j+1}))$
 - ii. if s_{j+1} is feasible then set $S_{j+1} \leftarrow S_{j+1} \cup \{s_{j+1}\}$
 - (b) $j \leftarrow j + 1$
4. return S_n

Fig. 6 DP()

To reduce the computational effort required by the dynamic programming approach, we can remove in each iteration some of the solutions in S_j if we know that one of the remaining solutions in S_j will eventually lead to an optimal solution if such a solution exists. A schedule $\bar{s} = (\bar{x}_i, \bar{y}_i, \bar{z}_i)_{1 \leq i \leq j}$ can be removed from S_j if another schedule $s = (x_i, y_i, z_i)_{1 \leq i \leq j}$ in S_j exists with

$$x_j^{\text{end}} \leq \bar{x}_j^{\text{end}}, \tag{14}$$

$$x_j^{\text{end}} - x_{k_s^c}^{\text{start}} - l_s^{\text{slack}|c} \leq \bar{x}_j^{\text{end}} - \bar{x}_{k_{\bar{s}}^c}^{\text{start}} - l_{\bar{s}}^{\text{slack}|c} \quad \text{for all } c \in C_1 \cup C_2, \tag{15}$$

$$x_j^{\text{end}} - x_{k_s^c}^{\text{start}} - l_s^{\text{push}|c} \leq \bar{x}_j^{\text{end}} - \bar{x}_{k_{\bar{s}}^c}^{\text{start}} - l_{\bar{s}}^{\text{push}|c} \quad \text{for all } c \in C_1 \cup C_2, \tag{16}$$

$$\sum_{i=k_s^c}^{j-1} d_{i,i+1} + \delta^c \sum_{i=k_s^c}^j w_i \leq \sum_{i=k_{\bar{s}}^c}^{j-1} d_{i,i+1} + \delta^c \sum_{i=k_{\bar{s}}^c}^j w_i \quad \text{for all } c \in C_3 \cup C_4, \tag{17}$$

and

$$x_j^{\text{end}} - x_1^{\text{start}} - l_s^{\text{slack}^*} \leq \bar{x}_j^{\text{end}} - \bar{x}_1^{\text{start}} - l_{\bar{s}}^{\text{slack}^*} \tag{18}$$

$$x_j^{\text{end}} - x_1^{\text{start}} - l_s^{\text{push}^*} \leq \bar{x}_j^{\text{end}} - \bar{x}_1^{\text{start}} - l_{\bar{s}}^{\text{push}^*} \tag{19}$$

Because of (15) and (16) and $\min\{l_s^{\text{slack}|c}, l_s^{\text{push}|c}\} = \min\{l_{\bar{s}}^{\text{slack}|c}, l_{\bar{s}}^{\text{push}|c}\} = 0$ we have $x_j^{\text{end}} - x_{k_c}^{\text{start}} \leq \bar{x}_j^{\text{end}} - \bar{x}_{k_c}^{\text{start}}$ for all $c \in C_1 \cup C_2$. Therefore, and because of (14) and (17) we know that, for given y_{j+1}, z_{j+1} , the solution obtained when extending s in the next iteration is feasible if the solution obtained when extending \bar{s} in the next iteration is feasible. Conditions (14) and (17) remain valid for the new solutions obtained. Furthermore, conditions (15) and (16) as well as conditions (18) and (19) remain valid because of condition (14). Therefore, the solution obtained by extending s can be transformed into a solution with the same or a smaller duration and conditions (14) to (19) remain valid. Thus, if \bar{s} can be extended to a feasible solution of the MD-TDSP, then s can be extended to a feasible solution of the MD-TDSP with the same or a smaller duration. Although we do not know whether the theoretical complexity of the algorithm can be reduced by removing unnecessary schedules, the computational experiments presented later show that the effort required by this approach is substantially smaller than the effort required by CPLEX 12 for solving the mixed integer program.

Extensions

It is obvious that the generic model presented in section “Problem formulation” cannot consider every rule that can possibly be imposed on truck driver schedules. Canadian and Australian hours of service regulations, for example, comprise additional constraints which are unique to these regulations. In Goel (2012a, b), it is shown how the model presented in this paper can be extended by introducing these additional constraints.

Although the model presented in section “Problem formulation” suffices to generate schedules complying with Regulation (EC) No. 561/2006, the optional rules of this regulation which allow taking break and rest periods in two parts are not considered. These rules give additional flexibility when generating truck driver schedules and can be easily considered in the dynamic programming approach.

According to the regulation, the first part of a rest period taken in two parts must have a duration of at least 3 h and the second part must have a duration of at least 9 h. The second part must be completed within 24 h after the end of the previous rest period. To consider this optional rule, we add a new rest type “shortrest” to R which represents the first part of a rest taken in two parts. We set $t^{\text{shortrest}} = 3$ and add “shortrest” to R^{nonstop} , because the short rest also resets the accumulated amount of driving without a regular break or rest. Furthermore, we modify the method $\text{extend}(\cdot)$ in such a way that after a short rest is scheduled (i.e. if $z_{j+1, \text{shortrest}} = 1$) then t^{rest} is set to 9 and t^{elapsed} is set to $24 - 9 = 15$. By this the time required for the next long rest period is appropriately reduced and the time that may elapse between two long rest periods is appropriately increased. After scheduling the next long rest (i.e. if $z_{j+1, \text{rest}} = 1$) the value of t^{rest} is reset to 11 and t^{elapsed} is reset to $24 - 11 = 13$. In certain cases, it may make sense to schedule a short rest immediately after a long

rest (see Goel 2010). Therefore, let us also add another rest type “rest-shortrest” to R and R^{nonstop} . The time required for taking a short rest immediately after a long rest is $t^{\text{rest-shortrest}}$ and is determined by $t^{\text{rest}} + t^{\text{shortrest}} = 14$. Within the method the values of t^{rest} , $t^{\text{rest-shortrest}}$, and t^{elapsed} are accordingly modified.

To appropriately modify the conditions used to remove schedules from a set S_j as described at the end of the previous section, let us include the modified values of t^r and t^c in the representation of a solution, i.e., we represent a schedule s by a tuple $((\bar{x}_i, \bar{y}_i, \bar{z}_i)_{1 \leq i \leq j}, (t^r)_{r \in R}, (t^c)_{c \in C})$. A schedule $\bar{s} = ((\bar{x}_i, \bar{y}_i, \bar{z}_i)_{1 \leq i \leq j}, (\bar{t}^r)_{r \in R}, (\bar{t}^c)_{c \in C})$ can be removed from S_j if another schedule $s = ((x_i, y_i, z_i)_{1 \leq i \leq j}, (t^r)_{r \in R}, (t^c)_{c \in C})$ in S_j exists with (14) to (19) and

$$t^r \leq \bar{t}^r \quad \text{for all } r \in R \tag{20}$$

and

$$t^c \geq \bar{t}^c \quad \text{for all } c \in C. \tag{21}$$

Condition (20) demands that the required duration of any type of rest $r \in R$ in s is not larger than in \bar{s} . Condition (21) demands that the limit given by $c \in C$ in s is not smaller than that in \bar{s} . If these conditions are satisfied we know that if \bar{s} can be extended to a feasible solution of the extension of the MD-TDSP, then s can be extended to a feasible solution with the same or a smaller duration.

The dynamic programming approach can be similarly extended to consider the rules allowing to take break periods in two parts. For this, an additional break type $t^{\text{shortbreak}}$ as well as other reasonable combinations of rests and breaks must be added and the method `extend(·)` must be appropriately modified.

Computational experiments

The approaches presented in this paper are evaluated on four randomly generated benchmark sets for a planning horizon starting on Monday 0.00 and ending on Friday 23.59. In all benchmark sets one hour of work time must be conducted at each work location in the tour and the driving time between two subsequent work locations is randomly set to a value between 1 and 10 h. Assuming an average speed of 75 km/h, this implies that the distance between two subsequent locations ranges from 75 to 750 km. Drivers may take rest periods before and after the work at any customer location. Furthermore, they may take rest periods at rest areas which are randomly distributed on the trip from one work location to another. The minimum driving time between rest areas is 15 min, and the maximum driving time 2 h. In the first benchmark set all locations have a single time window starting at some day in the planning horizon at 6.00 and ending at 20.00. In the second benchmark set, all locations have two time windows: the first starts at some day in the planning horizon at 6.00 and ends at 12.00, and the second starts at 14.00 and ends at 20.00. In the third and fourth benchmark sets, the time windows used in the first two benchmark sets are repeated on 2 days. Each benchmark set includes 1,000 instances. 769 these instances have an accumulated amount of driving and working of not more than

70 h and were used for the set of rules applicable in the United States. 471 of these instances have an accumulated amount of driving of not more than 56 h and an accumulated amount of driving and working of not more than 60 h and were used for the set of rules applicable in the European Union. The instances used for the rules applicable in the United States have between 27 and 81 customer locations and rest areas and the instances used for the rules applicable in the European Union have between 27 and 69 customer locations and rest areas.

In a first experiment, the impact of the restriction that rest periods must only be taken at customer locations or rest areas is made. For this purpose, the number of feasible solutions obtained for the problem studied in this paper is compared with the number of feasible solutions obtained for the truck driver scheduling problem in which rest periods may be taken anywhere. For this comparison we used the method for truck driver scheduling in the United States presented by Goel and Kok (2011) and the method for truck driver scheduling in the European Union presented by Goel (2010). As these methods only focus on determining feasible schedules, the duration of the schedules generated by these approaches is unnecessarily high. Table 1 gives the results of this experiments. In the table, the US hours of service regulations are indicated by “US”. The basic European Union regulations in which breaks and rest periods must not be split are indicated by “EU”. The extended dynamic programming approach for European Union regulations, in which the optional rules allowing to take breaks and rest periods in two parts are considered, is indicated by “EU*”. The additional restriction that rest periods must only be taken at customer locations or rest areas has a significant impact for all benchmark sets and a significant share of the instances for which a feasible truck driver schedule is found by the previous methods is infeasible if rest periods must only be taken at customer locations or rest areas. It is interesting to see that the restriction has a much larger impact in the European Union and more than a quarter of the truck driver schedules found is infeasible if rest periods must only be taken at customer locations or rest areas. With a larger number of breaks and rest periods which must be scheduled, the availability of suitable rest areas appears to be more important.

In a second experiment the performance of the methods presented in this paper is compared. Table 2 shows the results of the computational experiments conducted on a personal computer with an Intel 2.00 GHz CPU. In the table the average time required for solving an instance and the maximum computation time over all instances per benchmark set are reported. It can be seen that the dynamic programming approach requires significantly less average computation time than CPLEX 12 for all four benchmark sets and both the US and EU rules. Moreover, the maximum computation time required by the dynamic programming approach is also significantly smaller than the average computation time required by CPLEX 12, and the maximum computation time required by CPLEX 12 is prohibitively large if the problem is to be solved within an iterative approach for combined vehicle routing and truck driver scheduling. Not surprisingly, the fastest computation times are achieved for the first benchmark set in which all locations have a single time window. For US rules the dynamic programming approach only requires on average 1 ms, whereas CPLEX 12 requires more than two hundred times as much computation time. Interestingly, the average computation time required by CPLEX

Table 1 Number of instances for which a feasible solution can be found

Rules	Time windows	Instances	Anywhere	Rest areas
US	1 day: 6–20	769	468	419
US	1 day: 6–12, 14–20	769	455	412
US	2 days: 6–20	769	595	529
US	2 days: 6–12, 14–20	769	587	526
EU	1 day: 6–20	471	174	125
EU	1 day: 6–12, 14–20	471	174	124
EU	2 days: 6–20	471	228	159
EU	2 days: 6–12, 14–20	471	228	155
EU*	1 day: 6–20	471	177	134
EU*	1 day: 6–12, 14–20	471	177	129
EU*	2 days: 6–20	471	236	178
EU*	2 days: 6–12, 14–20	471	232	169

Table 2 Comparison of the approaches

Rules	Time windows	Instances	Feasible	Avg. duration (in min)	Avg. CPU DP (in ms)	Max. CPU DP (in ms)	Avg. CPU CPLEX 12 (in ms)	Max. CPU CPLEX 12 (in ms)
US	1 day: 6–20	769	419	5,600	1	4	227	747
US	1 day: 6–12, 14–20	769	412	5,626	2	5	238	769
US	2 days: 6–20	769	529	5,448	5	9	521	1,835
US	2 days: 6–12, 14–20	769	526	5,456	9	16	643	2,953
EU	1 day: 6–20	471	125	5,601	8	29	305	1,180
EU	1 day: 6–12, 14–20	471	124	5,595	13	50	317	1,211
EU	2 days: 6–20	471	159	5,603	36	74	577	4,533
EU	2 days: 6–12, 14–20	471	155	5,580	57	143	747	21,813
EU*	1 day: 6–20	471	134	5,552	14	51	–	–
EU*	1 day: 6–12, 14–20	471	129	5,523	37	157	–	–
EU*	2 days: 6–20	471	178	5,593	89	197	–	–
EU*	2 days: 6–12, 14–20	471	169	5,548	215	473	–	–

12 does not grow as much when changing the set of rules compared with the dynamic programming approach. However, the dynamic programming approach is still much faster than CPLEX 12 for the rules in the European Union. The computation times required by the dynamic programming approach for the extended

EU rules in which breaks and rests may be taken in two parts are a few times larger than those by the dynamic programming approach for the basic EU rules, but are still smaller than those required by CPLEX 12 for the basic EU rules. No results are reported for CPLEX 12 for the extended EU rules because it is yet not clear how the additional rules can be fully incorporated into a mixed integer program. An interesting result is that considering the possibility of taking breaks and rest periods in two parts not only helps in finding a truck driver schedule complying with EU regulations, but also allows to reduce schedule durations. The reason for this is that off-duty times which are not long enough for taking a long break or rest can be used to take short breaks and short rests. Therefore, otherwise unproductive off-duty times can be avoided and the driver requires less break and rest times later on.

To further evaluate the performance of the dynamic programming approach “EU*”, we conducted computational experiments on the instances used by Kok et al. (2011). These instances are obtained from known solutions of the vehicle routing problem with time windows by interpreting the time horizon as a period of 13 h. Within the planning horizon of 13 h no rest periods are required and the scheduling problem reduces to the problem of scheduling break periods and deciding whether these break periods are taken in two parts or not. Kok et al. (2011) report an average CPU time of 15 ms for these instances on a computer with a 2.83 GHz processor. The maximum CPU time required by the “EU*” approach was less than 2 ms, and thus remarkably smaller although a slower computer was used and all possible combinations of break and rest periods are evaluated by the “EU*” approach.

Conclusions

This paper studies the minimum duration truck driver scheduling problem (MD-TDSP), which is the problem of determining a schedule complying with applicable legislation in which all work activities begin within one of the corresponding time windows and which has the minimal duration. In the MD-TDSP, the truck driver may only take rest periods at customer locations or suitable rest areas. Most previous approaches for truck driver scheduling allowed truck drivers to take rest periods anywhere. The consideration of the locations of suitable rest areas, however, is particularly important if motorways are used and truck drivers must continue to drive until the next appropriate rest area is reached.

A generic model is presented which can be configured in such a way that the constraints of the current hours of service regulations in the United States, the new hours of service regulations in the United States which will become effective in 2013, and the basic set of rules imposed by European Union regulations are considered. Furthermore, the model can be used for other similarly structured regulations and provides a good starting point for regulations containing differently structured additional constraints.

A dynamic programming approach is presented which solves the MD-TDSP in a small fraction of time compared with the computation time required by one of the most advanced mixed integer programming solvers. Computational efficiency is

particularly important if the MD-TDSP is solved within iterative methods for solving vehicle routing and scheduling problems, where feasible truck driver schedules must be determined whenever the route of some vehicle is changed. The fast computation times are achieved by removing some of the partial schedules constructed during the course of the algorithm. If fast computation time is not required and if only those partial schedules are removed which cannot be extended to an optimal solution, many alternative optimal solutions for a given tour may be generated by the dynamic programming approach and truck drivers may be given the possibility of selecting one of the optimal rest patterns that fits best their individual preferences. Another important characteristic of the dynamic programming approach is that it can be easily adapted to consider additional rules not covered by the mixed integer programming formulation. This paper shows how the optional rules in the European Union which allow taking breaks and rest periods in two parts can be considered within the solution approach.

References

- Archetti C, Savelsbergh MWP (2009) The trip scheduling problem. *Transp Sci* 43(4):417–431
- Bartodziej P, Derigs U, Malcherek D, Vogel U (2009) Models and algorithms for solving combined vehicle and crew scheduling problems with rest constraints: an application to road feeder service planning in air cargo transportation. *OR Spectrum* 31:405–429
- Dasgupta S, Papadimitriou C, Vazirani U (2007) *Algorithms*. McGraw-Hill Higher Education, New York
- Derigs U, Kurowsky R, Vogel U (2011) Solving a real-world vehicle routing problem with multiple use of tractors and trailers and EU-regulations for drivers arising in air cargo road feeder services. *Eur J Oper Res* 213:309–319
- Drexl M, Prescott-Gagnon E (2010) Labelling algorithms for the elementary shortest path problem with resource constraints considering EU drivers' rules. *Logist Res* 2:79–96
- European Union (2002) Directive 2002/15/EC of the European Parliament and of the Council of 11 March 2002 on the organisation of the working time of persons performing mobile road transport activities. *Official J Eur Commun L* 80:35–39
- European Union (2006) Regulation (EC) No 561/2006 of the European Parliament and of the Council of 15 March 2006 on the harmonisation of certain social legislation relating to road transport and amending Council Regulations (EEC) No 3821/85 and (EC) No 2135/98 and repealing Council Regulation (EEC) No 3820/85. *Official Journal of the European Union L* 102, 11 April 2006
- European Union (2007) Opinion of the European Economic and Social Committee on the "European Road Safety Policy and Professional Drivers-Safe and secured parking places". *Official Journal of the European Union C* 175/88, 27 July 2007
- Federal Motor Carrier Safety Administration (2011) Hours of service of drivers. *Fed Reg* 76(248):81134–81188
- Goel A (2009) Vehicle scheduling and routing with drivers' working hours. *Transp Sci* 43(1):17–26
- Goel A (2010) Truck driver scheduling in the European Union. *Transp Sci* 44(4):429–441
- Goel A (2012a) A mixed integer programming formulation and effective cuts for minimising schedule durations of Australian truck drivers. *J Sched* doi:10.1007/s10951-012-0282-0
- Goel A (2012b) The Canadian minimum duration truck driver scheduling problem. *Comput Oper Res* 39(10):2359–2367
- Goel A (2012c) Truck driver scheduling in the United States and the 2013 rule change. Working paper, Zaragoza Logistics Center
- Goel A, Kok L (2011) Truck driver scheduling in the United States. *Transp Sci* 46(3):317–326
- Goel A, Rousseau LM (2011) Truck driver scheduling in Canada. *J Sched* (to appear)
- Goel A, Archetti C, Savelsbergh M (2012) Truck driver scheduling in Australia. *Comput Oper Res* 39(5):1122–1132

- Kok AL, Hans EW, Schutten WHM, Zijm MJM (2010a) A dynamic programming heuristic for vehicle routing with time-dependent travel times and required breaks. *Flexible Serv Manuf J* 22:83–108
- Kok L, Meyer CM, Kopfer H, Schutten MJM (2010b) A dynamic programming heuristic for the vehicle routing problem with time windows and European community social legislation. *Transp Sci* 44(4):442–454. doi:[10.1287/trsc.1100.0331](https://doi.org/10.1287/trsc.1100.0331)
- Kok AL, Hans EW, Schutten MJM (2011) Optimizing departure times in vehicle routes. *Eur J Oper Res* 210(3):579–587
- Meyer CM (2011) *Vehicle routing under consideration of driving and working hours: a distributed decision making perspective*. Gabler Verlag, Wiesbaden. ISBN:9783834929426
- National Transport Commission (2008a) Heavy vehicle driver fatigue reform: basic fatigue management explained. Information Bulletin. <http://www.ntc.gov.au/>
- National Transport Commission (2008b) Heavy vehicle driver fatigue reform: standard hours explained. Information Bulletin. <http://www.ntc.gov.au/>
- Prescott-Gagnon E, Desaulniers G, Drexel M, Rousseau LM (2010) European driver rules in vehicle routing with time windows. *Transp Sci* 44(4):455–473. doi:[10.1287/trsc.1100.0328](https://doi.org/10.1287/trsc.1100.0328)
- Rancourt ME, Cordeau JF, Laporte G (2010) Long-haul vehicle routing and scheduling with working hour rules. Technical Report CIRRELT-2010-46, CIRRELT. URL:<http://www.cirrelt.ca/DocumentsTravail/CIRRELT-2010-46.pdf>
- Transport Canada (2005) Commercial vehicle drivers hours of service regulations. SOR/2005-313. URL: <http://laws.justice.gc.ca/en/SOR-2005-313/>
- Xu H, Chen Z-L, Rajagopal S, Arunapuram S (2003) Solving a practical pickup and delivery problem. *Transp Sci* 37(3):347–364
- Zäpfel G, Bögl M (2008) Multi-period vehicle routing and crew scheduling with outsourcing options. *Int J Prod Econ* 113:980–996