*Manuscript
Click here to view linked References

Efficient Scheduling of Team Truck Drivers in the European Union

Asvin Goel[1,2] and Leendert Kok[3]

[1] MIT-Zaragoza International Logistics Program
Zaragoza Logistics Center, Spain
asvin@mit.edu

[2] Applied Telematics/e-Business Group,
Department of Computer Science, University of Leipzig, Germany
asvin.goel@uni-leipzig.de

[3] Operational Methods for Production and Logistics, University of Twente
P.O. Box 217, 7500AE, Enschede, Netherlands
a.l.kok@utwente.nl

November 24, 2009

Abstract

This paper studies the problem of scheduling working hours of team drivers in European road freight transport where a sequence of λ locations shall be visited within given time windows. Since April 2007 working hours of truck drivers in the European Union must comply with regulation (EC) No 561/2006. These regulations impose standard limits on the daily driving times of truck drivers and extended daily limits that may only be used twice a week for each driver. We present a depth-first-breadth-second search method which can find a feasible schedule complying with standard daily driving time limits in $O(\lambda^2)$ time, if such a schedule exists. Furthermore, we show that this method can also be used to find schedules complying with regulation (EC) No 561/2006 if daily driving times may exceed the standard limit.

1 Introduction

The European Transport Safety Council estimates that driver fatigue is a significant factor in approximately twenty per cent of commercial road transport crashes and reports that one out of two long haul drivers has fallen asleep while driving. To improve road safety and working conditions of drivers, the European Parliament and the Council of the European Union have adopted regulation (EC) No 561/2006 laying down provisions concerning driving and working hours of drivers in road transport. This regulation entered into force in April 2007 and applies to drivers of vehicles with a total mass of at least 3.5 tonnes and vehicles constructed to carry more than nine persons. According to the new regulation, road transport undertakings can be made liable for infringements committed by drivers. Neglecting regulatory constraints when scheduling driving and working hours of drivers may lead to infringements and/or delayed arrival times due to required breaks and rest periods that have not been scheduled. Consequently, road transport undertakings must ensure that truck driver schedules comply with regulation (EC) No 561/2006.

Scheduling working hours of truck drivers differs significantly from airline crew scheduling and driver scheduling in rail transport or mass transit systems which are covered by a comprehensive annotated bibliography by Ernst et al. (2004). The difference stems from the fact that in road freight transportation it is usually possible to interrupt transportation services in order to take compulsory breaks and rest periods. Furthermore, time constraints in road freight transport are usually not as strict and departure and arrival times can often be scheduled with some degree of freedom.

Perhaps the first work considering a combined vehicle routing and truck driver scheduling problem is the work by Savelsbergh and Sol (1998), who consider a problem in which lunch breaks and night breaks must be taken within fixed time intervals. The first work explicitly considering working hour regulations imposed by government agencies is the work of Xu et al. (2003). They conjecture that determining a minimal cost truck driver schedule complying with U.S. hours of service regulations is NP-hard in the presence of multiple time windows. Archetti and Savelsbergh (2009) present an algorithm for scheduling driving and working hours of truck drivers in the presence of single time windows and U.S. hours of service regulations. They prove that their algorithm finds a feasible truck driver schedule for a sequence of λ locations to be visited within single time windows in $O(\lambda^3)$ time if one exists. They conclude that the conjecture made by Xu et al. (2003) may not be true. Goel and Kok (2009) present more efficient algorithms which are able to find feasible truck driver schedules in $O(\lambda^2)$ time. Although restrictions imposed by U.S. hours of service regulations share some similarities with European legislation for team drivers, there are several differences. For example, European legislation requires that a new daily rest period is completed within 30 hours after the end of the last rest period. Furthermore, U.S. regulations do not allow to exceed the standard limit on the daily driving time. Thus, the methods developed by Archetti and Savelsbergh (2009) and Goel and Kok (2009) cannot be used for the European case.

European Union regulations for driving and working hours of truck drivers are receiving increasing attention because motor carriers and shippers can be made liable for infringement committed by the drivers. Goel (2009a), Kok et al. (2009), and Goel (2009c) present methods for generating truck driver schedules for vehicles manned by a single driver. Goel (2009b) studies the scheduling problem for team truck drivers in the European Union focussing on the standard driving time limit imposed by European regulation. Goel (2009b) identifies some structural properties of the truck driver scheduling problem and presents a depth-first search algorithm. This paper shows that by carefully exploring the search space the truck driver scheduling problem can be solved in $O(\lambda^2)$ time. Furthermore, we show that the case where team drivers may exceed standard daily driving time limits can be tackled without increasing the complexity.

The remainder of this paper is organised as follows. Section 2 describes the driving hour regulations for team drivers in the European Union. Section 3 introduces the truck driver scheduling problem and some definitions required throughout this paper. Section 4 presents an enumeration method for determining truck driver schedules in a so-called normal form. Section 5 presents a depth-first-breadth-second search method which can find a schedule complying with standard daily driving time limit in $O(\lambda^2)$ time if one exists. Section 6 shows that the depth-first-breadth-second search can also be used to find schedules complying with regulation (EC) No 561/2006 if standard daily driving time limits may be exceeded. Section 7 concludes this paper.

2 Regulation (EC) No 561/2006

This section describes relevant provisions of regulation (EC) No 561/2006 for vehicles continuously manned by two drivers. Regulation (EC) No 561/2006 distinguishes between four driver activities: rest periods, breaks, driving time, and other work. Rest periods are periods of at least 9 hours during which drivers may freely dispose of their time. During rest periods, the vehicle must remain stationary, implying that both drivers must take their rest period simultaneously. Breaks are short periods exclusively used for recuperation, during which a driver may not carry out any work. During break periods, the vehicle does not have to remain stationary and one driver may take a break while the other is driving. 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.

According to regulation (EC) No 561/2006, a daily rest periods of 9 hours must be completed within 30 hours after the end of the previous rest period. The standard set of rules constrains the accumulated driving time of each driver between two consecutive rest periods to at most 9 hours. Thus, the accumulated driving time of both drivers is limited to 18 hours. Twice a week, however, each driver may drive up to at most ten hours between two consecutive rest periods. Thus, the accumulated driving time of both drivers is limited to 18 hours if no driver makes use of extended driving times, to 19 hours if one of the drivers makes use of extended driving times, and to 20 hours if both drivers make use of extended driving times.

Figure 1: Driving times, breaks, and rest periods for vehicles manned by two drivers

As illustrated in Figure 1, regulation (EC) No 561/2006 has a big influence on total travel times since a big part of total travel times results from periods in which the vehicle is not moving. In this paper, we consider a planning horizon of one week and assume that limits on the weekly amount of driving and working are complied with. A full description of the regulation including limits on the weekly amount of driving and working can be found in European Union (2006).

3 The Truck Driver Scheduling Problem

Let us consider a sequence of locations denoted by $n_1, n_2, \dots, n_\lambda$ which shall be visited by a double manned vehicle. At each location n_μ some stationary work of duration w_{n_μ} shall be conducted. This work shall begin within a time window denoted by T_{n_μ}. We assume that n_1 corresponds to the vehicle's location at time zero and that drivers complete their work week after finishing work at location n_λ. The driving time required for moving from node n_μ to node $n_{\mu+1}$ shall be denoted by $\delta_{\mu,\mu+1}$.

The truck driver scheduling problem is the problem of scheduling driving, working, and rest periods in such a way that all locations are visited within the given time windows and that driving and working hours of the truck drivers comply with applicable legislation.

Let us denote with `DRIVE` any period of consecutive driving, with `WORK` any period of work during which the drivers are not driving, with `REST` any rest period of at least 9 hours, and with `IDLE` any other period which is neither regarded as driving, work, or rest period.

Throughout this paper, we assume that at the beginning of the planning horizon both drivers return from a rest period long enough, such that previous activities have no effect on the amount of driving and working within the planning horizon. In the case where drivers must not take extended daily driving times, the following parameters are given by the regulation:

- The minimum duration of a rest period is $t^{\text{rest}} := 9$ hours
- The maximum time after the end of the previous rest period after which a new daily rest period must be completed is $t^{\text{day}} := 30$ hours
- The maximum accumulated driving time of both drivers between two rest periods is $t^{\text{drive}} :=$ 18 hours

The general case in which daily driving times may exceed the standard limits is covered in Section 6.

A schedule can be specified by a sequence of activities to be performed by the drivers. Let $\mathcal{A} := \{a = (a^{\text{type}}, a^{\text{length}}) \mid a^{\text{type}} \in \{\texttt{DRIVE}, \texttt{WORK}, \texttt{REST}, \texttt{IDLE}\}, a^{\text{length}} > 0\}$ denote the set of activities that may be scheduled. Let « . » be an operator that concatenates different activities. Thus, $a_1.a_2. \ \ldots \ .a_k$ denotes a *schedule* in which for each $i \in \{1, 2, \ldots, k-1\}$ activity a_{i+1} is performed immediately after activity a_i. For a given schedule $s := a_1.a_2. \ \ldots \ .a_k$ and $i, j \in \{1, 2, \ldots, k\}$ let $s_{i,j} := a_i.a_{i+1}. \ \ldots \ .a_j$ denote the partial schedule composed of activities a_i to a_j. Unless otherwise stated, in the remainder of this paper, we assume that a_1 to a_k denote the activities of a schedule s, i.e. we assume that $s = a_1.a_2. \ \ldots \ .a_k$.

For the ease of notation and without loss of generality, we assume that each schedule begins with a rest period and that $\min T_{n_\mu} > t^{\text{rest}}$. Let us now define some properties of a schedule $s = a_1.a_2. \ \ldots \ .a_k$. For this, let $i := \max\{i' \mid 1 \leq i' \leq k, a_{i'}^{\text{type}} = \texttt{REST}\}$ denote the index of the last rest period in the schedule.

- The completion time of the work plan is

$$l_s^{\text{end}} := \sum_{1 \leq j \leq k} a_j^{\text{length}}$$

- The accumulated driving time since the last rest period is

$$l_s^{\text{daily}} := \sum_{\substack{i < j \leq k \\ a_j^{\text{type}} = \texttt{DRIVE}}} a_j^{\text{length}}$$

- The time of completion of the last daily rest period is

$$l_s^{\text{last\_rest}} := l_{s_{1,i}}^{\text{end}}$$

With this notation we can now give a definition for feasible truck driver schedules for a tour $\theta := (n_1, \ldots, n_\lambda)$. Let us consider a sequence of activities $a_1, a_2, \ldots, a_k$ which contains exactly λ stationary work periods. Let us denote with $i(\mu)$ the index corresponding to the μth stationary work period, i.e. $a_{i(\mu)}$ corresponds to work performed at location n_μ. For any $1 \leq i \leq k$ with $a_i^{\text{type}} = \texttt{WORK}$ let us denote with $n(i)$ the respective location in tour θ, i.e. $n(i(\mu)) = n_\mu$.

Definition A schedule $s = a_1.a_2.\ \dots\ .a_k$ is a *feasible schedule* for tour $\theta := (n_1, n_2, \dots, n_\lambda)$ if and only if

$$\sum_{\substack{1\leq j\leq k\\ a_j^{\text{type}}=\texttt{WORK}}} 1 = \lambda \quad \text{and} \sum_{\substack{i(1)\leq j<i(\lambda)\\ a_j^{\text{type}}=\texttt{DRIVE}}} a_j^{\text{length}} = \sum_{\substack{1\leq j\leq k\\ a_j^{\text{type}}=\texttt{DRIVE}}} a_j^{\text{length}} \tag{1}$$

$$a_{i(\mu)}^{\text{length}} = w_{n_\mu} \text{ for each } \mu \in \{1, 2, \dots, \lambda\} \tag{2}$$

$$l_{s_{1,i(\mu)-1}}^{\text{end}} \in T_{n_\mu} \text{ for each } \mu \in \{1, 2, \dots, \lambda\} \tag{3}$$

$$\sum_{\substack{i(\mu)<j<i(\mu+1)\\ a_i^{\text{type}}=\texttt{DRIVE}}} a_j^{\text{length}} = \delta_{\mu,\mu+1} \text{ for each } \mu \in \{1, 2, \dots, \lambda-1\} \tag{4}$$

$$l_{s_{1,i}}^{\text{drive}} \leq t^{\text{drive}} \text{ for each } 1 \leq i \leq k \tag{5}$$

$$a_i^{\text{length}} \geq t^{\text{rest}} \text{ for each } 1 \leq i \leq k \text{ with } a_i^{\text{type}} = \texttt{REST} \tag{6}$$

$$l_{s_{1,i}}^{\text{end}} + t^{\text{rest}} \leq l_{s_{1,i}}^{\text{last\_rest}} + t^{\text{day}} \text{ for each } 1 \leq i \leq k \text{ with } a_i^{\text{type}} \neq \texttt{REST} \tag{7}$$

Condition (1) demands that the work plan contains exactly λ stationary work periods and that all driving activities are scheduled between the first and the last stationary work period. Condition (2) demands that the duration of the μth work activity matches the specified work duration at location n_μ. Condition (3) demands that each work activity begins within the given time window. Condition (4) demands that the accumulated driving time between two work activities matches the driving time required to move from one location to the other. Conditions (5) and (6) guarantee that daily driving time limits are not exceeded and that the duration of rest periods is long enough to start a new working day. Condition (7) guarantees that a rest period of at least 9 hours is completed 30 hours after the end of the previous rest period.

Because of condition (7) it may be required that rest periods of more than 9 hours duration have to be taken. Otherwise, the time between the completion of two subsequent rest periods may exceed the 30 hour limit. Let us now introduce the concept of *pseudo-feasibility* which allows us to restrict our search to schedules in which each rest period has a duration of t^{rest}. For any partial schedule $s_{i,j} = a_i.a_{i+1}.\ \dots\ .a_j$ let

$$l_{s_{i,j}}^{\text{slack}} := \sum_{\substack{i\leq i'\leq j\\ a_{i'}^{\text{type}}=\texttt{IDLE}}} a_{i'}^{\text{length}}$$

denote the accumulated slack time. Given a schedule $s = a_1.a_2.\ \dots\ .a_k$ and some $1 \leq i \leq k$ with $a_i^{\text{type}} = \texttt{REST}$, we know that $l_{s_{i,k}}^{\text{slack}}$ is the maximum amount of time by which the rest period may be extended without increasing the completion time. For each $1 < i \leq k$ with $a_i^{\text{type}} = \texttt{WORK}$, we know that $\max T_{n(i)} - l_{s_{1,i-1}}^{\text{end}}$ is the maximum amount by which the start of the work period can be postponed without violating time window constraints. Thus, the maximum amount by which the duration of the last rest period in schedule s may be extended without increasing the completion time or violating time window constraints is

$$l_s^{\text{extend}} := \min\left\{l_{s_{i,k}}^{\text{slack}},\ \min_{\substack{i<j\leq k\\ a_j^{\text{type}}=\texttt{WORK}}} \left\{l_{s_{i,j}}^{\text{slack}} + \max T_{n(j)} - l_{s_{1,j-1}}^{\text{end}}\right\}\right\}$$

where $i = \max\left\{i' \mid 1 \leq i' \leq k, a_{i'}^{\text{type}} = \texttt{REST}\right\}$. We can now define criteria for pseudo-feasible schedules.

Definition The schedule $s = a_1.a_2.\ \dots\ .a_k$ is a *pseudo-feasible schedule* for tour θ if and only if (1) - (6) and

$$l_{s_{1,i}}^{\text{end}} + t^{\text{rest}} \leq l_{s_{1,i}}^{\text{last\_rest}} + l_{s_{1,i}}^{\text{extend}} + t^{\text{day}} \text{ for each } 1 \leq i \leq k \text{ with } a_i^{\text{type}} \neq \texttt{REST} \tag{7'}$$

Goel (2009b) shows that the problem of finding a feasible schedule for a given tour can be replaced by the problem of finding a pseudo-feasible schedule for the tour, and that each pseudo-feasible schedule for a tour θ can be transformed into a feasible schedule for tour θ. The normal form for schedules provided by Goel (2009b) allows us to restrict the search space in such a way that it can be efficiently explored.

Definition A pseudo-feasible schedule $s = a_1.a_2.\ \dots\ .a_k$ for a tour θ is in *normal form* if and only if for each $1 \leq i \leq k$

$$a_i^{\text{type}} = a_{i+1}^{\text{type}} \Rightarrow a_i^{\text{type}} = a_{i+1}^{\text{type}} = \texttt{WORK} \tag{N1}$$

$$a_i^{\text{type}} = \texttt{IDLE} \Rightarrow a_{i+1}^{\text{type}} = \texttt{WORK} \tag{N2}$$

$$a_i^{\text{type}} = \texttt{REST} \text{ and } a_{i+1}^{\text{type}} = \texttt{DRIVE} \ \Rightarrow\ \begin{array}{l} l_{s_{1,i-1}}^{\text{drive}} = t^{\text{drive}} \text{ or} \\ l_{s_{1,i-1}}^{\text{end}} = l_{s_{1,i-1}}^{\text{last\_rest}} + l_{s_{1,i-1}}^{\text{extend}} + t^{\text{day}} \end{array} \tag{N3}$$

$$a_i^{\text{type}} = \texttt{IDLE} \Rightarrow a_i^{\text{length}} = \min T_{n(i+1)} - l_{s_{1,i-1}}^{\text{end}} \tag{N4}$$

$$a_i^{\text{type}} = \texttt{REST} \Rightarrow a_i^{\text{length}} = t^{\text{rest}} \tag{N5}$$

(N1) demands that two consecutive activities are of different type unless they are working activities. If (N1) is violated we can simply merge two consecutive activities of the same type. (N2) demands that idle periods are only scheduled immediately before work periods. If (N2) is violated we can switch the positions of the idle period and the following activity. (N3) demands that a rest period which is followed by a driving period is only scheduled if no additional driving activity could have been scheduled before the rest. If (N3) is violated we can schedule a part of the following driving activity before the rest. (N4) and (N5) demand that the duration of all idle and rest periods is as short as possible. If either of the conditions is violated we can reduce the length of that period.

Goel (2009b) shows that each pseudo-feasible schedule can be converted into a pseudo-feasible schedule in normal form. Thus, the truck driver scheduling problem can be reduced to the problem of finding pseudo-feasible schedules in normal form.

4 Enumeration

This section presents a method for determining all pseudo-feasible schedules in normal form. The trip calculation method illustrated in Figure 2 can be used to determine the activities to be performed to reach the next destination. The trip calculation first determines the maximum amount of driving until either the next destination is reached or a rest period must be scheduled. Then a driving activity is appended to the schedule. If the next destination is not yet reached a rest period is scheduled and the trip calculation method determines the duration of the next driving period. This process is continued until the next destination is reached.

We can now enumerate all pseudo-feasible schedules in normal form as follows. Let us set $\mathcal{A}_{n_1} := \emptyset$ and $\mathcal{B}_{n_1} := \{(\texttt{REST}, t^{\text{rest}}).(\texttt{IDLE}, \min T_{n_1} - t^{\text{rest}}).(\texttt{WORK}, w_{n_1})\}$. For any partial schedule s let us denote with $\rho(s)$ the output of the trip calculation method illustrated in Figure 2. For each $1 < \mu \leq \lambda$ let

$$\begin{aligned}\hat{\mathcal{A}}_{n_{\mu+1}} := &\left\{ \rho(s).(\texttt{WORK}, w_{n_{\mu+1}}) \mid s \in \mathcal{A}_{n_\mu} \cup \mathcal{B}_{n_\mu}, l^{\text{end}}_{\rho(s)} \in T_{n_{\mu+1}} \right\} \cup \\ &\left\{ \rho(s).(\texttt{IDLE}, \min T_{n_{\mu+1}} - l^{\text{end}}_{\rho(s)}).(\texttt{WORK}, w_{n_{\mu+1}}) \mid s \in \mathcal{A}_{n_\mu} \cup \mathcal{B}_{n_\mu}, l^{\text{end}}_{\rho(s)} < \min T_{n_{\mu+1}} \right\}\end{aligned}$$

and

$$\mathcal{A}_{n_{\mu+1}} := \{ s \in \hat{\mathcal{A}}_{n_{\mu+1}} \mid l^{\text{end}}_s + t^{\text{rest}} \leq l^{\text{last\_rest}}_s + l^{\text{extend}}_s + t^{\text{day}} \}$$

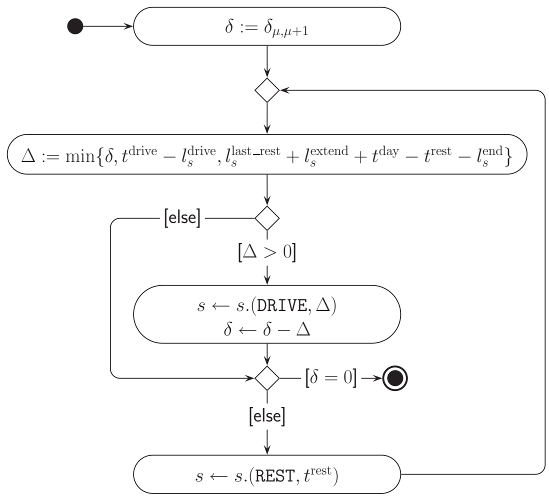

Figure 2: Trip calculation method

and

$$\hat{\mathcal{B}}_{n_{\mu+1}} := \big\{\, \rho(s).(\texttt{REST}, t^{\text{rest}}).(\texttt{WORK}, w_{n_{\mu+1}}) \mid s \in \mathcal{A}_{n_\mu} \cup \mathcal{B}_{n_\mu}, l^{\text{end}}_{\rho(s)} + t^{\text{rest}} \in T_{n_{\mu+1}} \big\} \cup \big\{\, \rho(s).(\texttt{REST}, t^{\text{rest}}).(\texttt{IDLE}, \min T_{n_{\mu+1}} - l^{\text{end}}_{\rho(s)} - t^{\text{rest}}).(\texttt{WORK}, w_{n_{\mu+1}}) \mid s \in \mathcal{A}_{n_\mu} \cup \mathcal{B}_{n_\mu}, l^{\text{end}}_{\rho(s)} + t^{\text{rest}} < \min T_{n_{\mu+1}} \big\}.$$

and

$$\mathcal{B}_{n_{\mu+1}} := \{ s \in \hat{\mathcal{B}}_{n_{\mu+1}} \mid l_s^{\text{end}} + t^{\text{rest}} \le l_s^{\text{last\_rest}} + l_s^{\text{extend}} + t^{\text{day}} \}$$

Lemma 1 For each $1 \le \mu \le \lambda$ the set of all pseudo-feasible schedules in normal form for tour $\theta_\mu := (n_1, \ldots, n_\mu)$ is $\mathcal{A}_{n_\mu} \cup \mathcal{B}_{n_\mu}$.

Proof. For $\mu = 1$ the schedule $s \in \mathcal{B}_{n_1}$ is the only pseudo-feasible schedule in normal form. Assume the statement is true for some $\mu < \lambda$. We show that the statement holds for $\mu \leftarrow \mu + 1$. First, note that each pseudo-feasible schedule in normal form for $\theta_{\mu+1}$ must be an extension of a pseudo-feasible schedule in normal form for θ_μ. Thus, $\mathcal{A}_{n_{\mu+1}} \cup \mathcal{B}_{n_{\mu+1}}$ only contains feasible extensions of schedules in $\mathcal{A}_{n_\mu} \cup \mathcal{B}_{n_\mu}$. Because of the normality conditions we know that idle periods are taken as late and as short as possible and that driving periods are taken as early as possible. Rest periods are only taken when required to continue driving or after arriving at the next customer location.

Thus, any pseudo-feasible schedule in normal form for tour $\theta_{\mu+1}$ which starts with a partial schedule $s \in \mathcal{A}_{n_\mu} \cup \mathcal{B}_{n_\mu}$ must continue with the activities determined by the trip calculation method until location $n_{\mu+1}$ is reached. If $l^{\text{end}}_{\rho(s)} < \min T_{n_{\mu+1}}$ then $\rho(s)$ must continue with an idle period of minimum duration or a rest period. In the latter case, the rest period may be succeeded by an idle period of minimum length until the time window opens. After that, the next work period must be scheduled. If $l^{\text{end}}_{\rho(s)} \geq \min T_{n_{\mu+1}}$ then $\rho(s)$ must continue with the next work period or a rest period. In the latter case, the rest period must be succeeded by the next work period. Thus, $\mathcal{A}_{n_{\mu+1}} \cup \mathcal{B}_{n_{\mu+1}}$ contains all pseudo-feasible schedules in normal form for tour $\theta_{\mu+1}$. By definition all schedules in $\mathcal{A}_{n_{\mu+1}} \cup \mathcal{B}_{n_{\mu+1}}$ are pseudo-feasible and in normal form. □

Generating all pseudo-feasible schedules in normal form results in a search tree with up to $2^{\lambda-1}$ leaves. In general, it is not efficient to completely enumerate the tree. In order to efficiently solve the truck driver scheduling problem we need to constrain the number of schedules which may be part of a solution. Let $\mathcal{S}(\theta)$ denote the set of pseudo-feasible schedules for a tour $\theta := (n_1, n_2, \ldots, n_\lambda)$ and for any schedule s let

$$\mathcal{S}(\theta, s) := \{\hat{s} \mid s.\hat{s} \in \mathcal{S}(\theta)\}$$

denote the set of schedules $\hat{s}$ for which $s.\hat{s}$ is a pseudo-feasible schedule for tour θ.

Definition Let $\theta := (n_1, n_2, \ldots, n_\lambda)$ and for each $1 \leq \mu \leq \lambda$ let $\theta_\mu := (n_1, n_2, \ldots, n_\mu)$. A schedule $s' \in \mathcal{S}(\theta_\mu)$ *dominates* $s'' \in \mathcal{S}(\theta_\mu)$ if some schedule $\tilde{s}$ exists such that

$$\tilde{s}.\hat{s} \in \mathcal{S}(\theta, s') \text{ for all } \hat{s} \in \mathcal{S}(\theta, s'').$$

Thus, if s'' is dominated by s' we know that for any $\hat{s}$ for which $s''.\hat{s}$ is a pseudo-feasible schedule for tour θ there exists a schedule $\tilde{s}$ for which $s'.\tilde{s}.\hat{s}$ is a pseudo-feasible schedule for tour θ. Thus, it suffices to search for schedules in $\mathcal{S}(\theta, s')$ and we do not need to consider the dominated schedule s'' in our search for a feasible schedule for θ.

Lemma 2 Schedule $s' \in \mathcal{S}(\theta_\mu)$ dominates $s'' \in \mathcal{S}(\theta_\mu)$ if

$$l^{\text{end}}_{s'} \leq l^{\text{end}}_{s''} \text{ and } l^{\text{drive}}_{s'} \leq l^{\text{drive}}_{s''} \text{ and } l^{\text{last\_rest}}_{s'} + l^{\text{extend}}_{s'} \geq l^{\text{last\_rest}}_{s''} + l^{\text{extend}}_{s''}$$

Proof. $\tilde{s} := (\texttt{IDLE}, l^{\text{end}}_{s''} - l^{\text{end}}_{s'})$ if $l^{\text{end}}_{s'} < l^{\text{end}}_{s''}$, and $\tilde{s} := \emptyset$ otherwise. □

Lemma 3 A schedule $s' \in \mathcal{B}_{n_\mu}$ dominates all other schedules $s'' \in \mathcal{A}_{n_\mu} \cup \mathcal{B}_{n_\mu}$ with $l_{s'}^{\text{end}} \leq l_{s''}^{\text{end}}$.

Proof. For each $s' \in \mathcal{B}_{n_\mu}$ we have $l_{s'}^{\text{drive}} = 0$ and $l_{s'}^{\text{last\_rest}} + l_{s'}^{\text{extend}} = l_{s'}^{\text{end}} - w_{n_\mu}$. If $l_{s'}^{\text{end}} = l_{s''}^{\text{end}}$ then conditions of Lemma 2 are satisfied. Otherwise, $\tilde{s} := (\texttt{IDLE}, l_{s''}^{\text{end}} - l_{s'}^{\text{end}})$. □

With these dominance criteria we do not need to enumerate the complete search tree because many branches will not be required in our search for a feasible schedule.

5 Depth-First-Breadth-Second Search

In this section we present a depth-first-breadth-second search (DFBSS) algorithm which uses the dominance criteria given in the previous section to cut off branches corresponding to dominated schedules. Figure 3 illustrates this algorithm. The DFBSS begins with initialising for each $1 \leq \mu \leq \lambda$ the set $\mathcal{S}_{n_\mu}$ of already known pseudo-feasible schedules in normal form for tour θ_μ, and the set $\mathcal{S}^*_{n_\mu}$ of schedules for tour θ_μ which have already been selected as input for the trip calculation method. The set $\mathcal{S}_{n_1}$ is uniquely determined. All other sets are empty at the beginning of the algorithm. The algorithm begins iterating with $\mu = 1$. Among all schedules in $\mathcal{S}_{n_\mu}$ which have not yet been selected the algorithm selects the schedule s with earliest completion time. This schedule is included to the set $\mathcal{S}^*_{n_\mu}$ of already selected schedules and used by the trip calculation method. The set $\mathcal{S}_{n_{\mu+1}}$ of known solutions for tour $\theta_{\mu+1}$ is now updated by adding up to two schedules obtained by taking the solution $\rho(s)$ and adding a work period at the earliest possible time or adding a work period at the earliest possible time after completion of an additional rest period. After this update some of the schedules in $\mathcal{S}_{n_{\mu+1}}$ may be dominated and can be removed. The DFBSS increments μ and continues with the next schedule in $\mathcal{S}_{n_\mu} \setminus \mathcal{S}^*_{n_\mu}$. If no such schedule exists, μ is reduced to the smallest value for which such a schedule exists. The algorithm terminates when the first schedule in $\mathcal{S}_{n_\lambda}$ is found or $\mu = \lambda$.

In order to find an upper bound on the number of iterations performed by the DFBSS let us first give an upper bound on the number of schedules in $\mathcal{S}_{n_{\mu'}} \setminus \mathcal{S}^*_{n_{\mu'}}$ for each $1 < \mu' \leq \lambda$, i.e. the number of schedules determined by the search whose branches are not yet explored when traversing the reference point in Figure 3.

Lemma 4 At the reference point in Figure 3 we have $|\mathcal{S}_{n_{\mu'}} \setminus \mathcal{S}^*_{n_{\mu'}}| \leq 1$ for each $1 < \mu' \leq \lambda$.

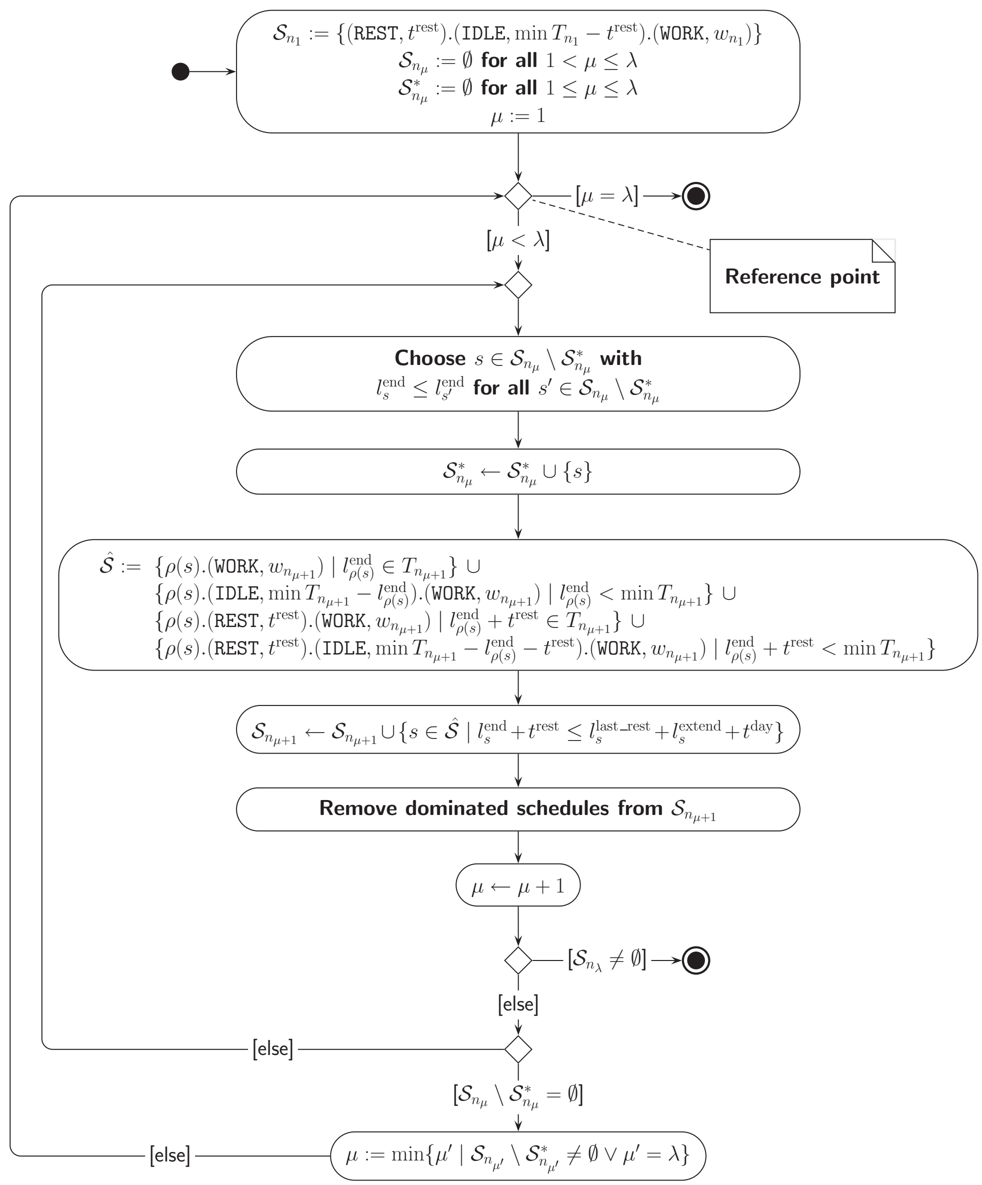

Figure 3: Depth-first-breadth-second search

Proof. We will first show another property of the DFBSS. In the beginning of the DFBSS we have $\mathcal{A}_{n_{\mu'}} \cap \mathcal{S}_{n_{\mu'}} \setminus \mathcal{S}^*_{n_{\mu'}} = \emptyset$ for each $1 < \mu' \leq \lambda$. In each iteration of the search at most two new schedules are included into the set $\mathcal{S}_{n_{\mu+1}}$. At most one of these two schedules is in $\mathcal{A}_{n_{\mu+1}}$ and at most one is in $\mathcal{B}_{n_{\mu+1}}$. If $\mathcal{S}_{n_{\mu+1}} \setminus \mathcal{S}^*_{n_{\mu+1}} \neq \emptyset$ either the new schedule in $\mathcal{A}_{n_{\mu+1}}$ or a schedule in $\mathcal{B}_{n_{\mu+1}} \cap \mathcal{S}_{n_{\mu+1}}$ is chosen in the next step and inserted to $\mathcal{S}^*_{n_{\mu+1}}$. If a schedule in $\mathcal{B}_{n_{\mu+1}} \cap \mathcal{S}_{n_{\mu+1}}$ is chosen, either no schedule in $\mathcal{A}_{n_{\mu+1}}$ was included into the set $\mathcal{S}_{n_{\mu+1}}$ or the schedule is dominated by the chosen schedule and removed. Thus, each time a new schedule in $\mathcal{A}_{n_{\mu+1}}$ is included in $\mathcal{S}_{n_{\mu+1}}$ it is either included into $\mathcal{S}^*_{n_{\mu+1}}$ in the next iteration or immediately removed because the schedule is dominated by a schedule in $\mathcal{B}_{n_{\mu+1}}$. Therefore, each time the reference point is passed $\mathcal{A}_{n_{\mu'}} \cap \mathcal{S}_{n_{\mu'}} \setminus \mathcal{S}^*_{n_{\mu'}} = \emptyset$ for each $1 < \mu' \leq \lambda$.

Because of Lemma 3 we know that after removing dominated schedules $|\mathcal{B}_{n_{\mu'}} \cap \mathcal{S}_{n_{\mu'}} \setminus \mathcal{S}^*_{n_{\mu'}}| \leq 1$ for each $1 < \mu' \leq \lambda$. Thus, combining these two properties we know that each time the reference point is passed $|\mathcal{S}_{n_{\mu'}} \setminus \mathcal{S}^*_{n_{\mu'}}| \leq 1$ for each $1 < \mu' \leq \lambda$. □

Lemma 5 The DFBSS algorithm terminates after at most $\frac{1}{2}\lambda^2 - \frac{1}{2}\lambda$ iterations.

Proof. Let μ^i denote the value of μ at the ith time the reference point is passed. The search iterates at most $\lambda - \mu^i$ times before passing the reference point again. Because of Lemma 4 we know that each time after passing the reference point the only schedule in $\mathcal{S}_{n_{\mu^i}} \setminus \mathcal{S}^*_{n_{\mu^i}}$ is chosen and included to $\mathcal{S}^*_{n_{\mu^i}}$. Thus, $\mu^i < \mu^{i+1}$ and the DFBSS algorithm terminates after at most

$$\sum_{1 \leq \mu < \lambda} (\lambda - \mu) = \frac{\lambda(\lambda - 1)}{2}$$

iterations. □

6 Extended Daily Driving Times

So far we have only considered the standard limit on daily driving times. This section shows how the method presented in the previous section can be used for the case where the combined daily driving time may exceed the standard limit. Twice a week each driver may exceed the standard daily driving time of 9 hours and may drive for up to at most 10 hours. As we only consider a planning horizon of one week, at most four extra hours may be taken throughout the week. As each driver may only drive up to nine plus one hour without a rest period, the maximum number of extra hours taken between two subsequent rest periods is at most two.

The accumulated driving time between any two rest periods can add up to 18 hours or more at most six times within a planning horizon of at one week (168 hours), because less than 18 hours of driving can be performed after $6 \times (18 + 9) = 162$ hours since the beginning of the planning horizon. Each time the accumulated daily driving time reaches 18 hours we can decide whether the daily driving time of one or both drivers may be extended. Let

$$K := \Big\{ (\kappa_1, \kappa_2, ..., \kappa_6) \in \{0, 1, 2\}^6 \mid \sum_{1 \leq i \leq 6} \kappa_i = 4 \Big\}$$

denote a configuration corresponding to all the schedules complying with the regulation in which κ_i denotes whether the daily driving time of no driver ($\kappa_i = 0$), one driver ($\kappa_i = 1$), or both drivers ($\kappa_i = 2$) may be extended at the ith time the accumulated daily driving time reaches 18 hours. The number of different configurations K equals 90 because there are 15 configurations in which 4 different daily driving times may be extended, 60 configurations in which one daily driving time may be extended twice and three others may be extended once, and 15 configurations in which two daily driving times may be extended twice. For each of these configurations K, we can use the modified trip calculation method illustrated in Figure 4.

The modified trip calculation begins with initialising δ representing the remaining driving time required to reach the next location, and i indicating the number of rest periods in schedule s which follow a daily driving time of 18 hours or more. Then, the algorithm determines the duration of the next driving period based on the daily driving time limit increased by κ_{i+1}. The modified trip calculation continues with the same steps as in the original method and updates i in each iteration.

As there is a constant bound on the number of configurations, we retain a complexity of $O(\lambda^2)$ for solving the truck driver scheduling problem with extended daily driving times. It must be noted that it is not necessary to perform all the steps of the DFBSS algorithm for each configuration. Instead, we can eliminate redundant steps for similar configurations and reduce the computational effort accordingly.

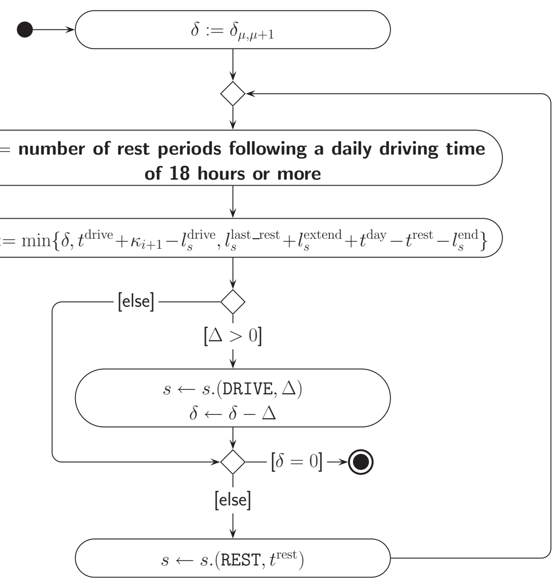

Figure 4: Modified trip calculation method

7 Conclusions and Final Remarks

Scheduling of working and driving hours of truck drivers is a complex task in the presence of legislatory constraints. Despite their importance, restrictions on drivers' working hours have only attracted very little interest in the literature. This paper studies truck driver scheduling problems considering the European Union regulations for team drivers. A depth-first-breadth-second search algorithm is presented, which can find a schedule complying with standard daily driving time limits in $O(\lambda^2)$ time if one exists. We furthermore show, that the general case in which standard driving time limits may be exceeded, can be solved without increasing the complexity.

References

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

A. T. Ernst, H. Jiang, M. Krishnamoorthy, B. Owens, and D. Sier. An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, 127(1):21–144, March 2004. doi: 10.1023/b:anor.0000019087.46656.e2.

European Transport Safety Council. The role of driver fatigue in commercial road transport crashes. ETSC Review, 2001. URL `www.etsc.eu`.

European Union. 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, 11.04.2006, 2006.

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

A. Goel. Scheduling of Working Hours of Team Drivers in European Road Transport. Working Paper, University of Leipzig, 2009b.

A. Goel. Truck Driver Scheduling and Regulation (EC) No 561/2006. Working Paper, MIT-Zaragoza International Logistics Program, 2009c.

A. Goel and A. L. Kok. Efficient Truck Driver Scheduling in the United States. Working Paper, University of Leipzig, 2009.

A. L. Kok, C. M. Meyer, H. Kopfer, and J. M. J. Schutten. Dynamic Programming Algorithm for the Vehicle Routing Problem with Time Windows and EC Social Legislation. Beta working paper 270, University of Twente, 2009.

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

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