Truck Driver Scheduling in Canada

Asvin Goel[1,2], Louis-Martin Rousseau[3]

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

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

[3] École Polytechnique de Montréal and CIRRELT,
Montréal, Québec H3C 3AT, Canada, louis-martin.rousseau@cirrelt.ca

December 8, 2010

Abstract

This paper presents and studies the Canadian Truck Driver Scheduling Problem (CAN-TDSP) which is the problem of determining whether a sequence of locations can be visited within given time windows in such a way that driving and working activities of truck drivers comply with Canadian Commercial Vehicle Drivers Hours of Service Regulations. Canadian regulations comprise the provisions found in U.S. hours of service regulations as well as additional constraints on the maximum amount of driving and the minimum amount of off-duty time on each day. We present two heuristics and an exact approach for solving the CAN-TDSP. Computational experiments demonstrate the effectiveness of our approaches and indicate that Canadian regulations are significantly more permissive than U.S. hours of service regulations.

1 Introduction

According to European Transport Safety Council (2001), Federal Motor Carrier Safety Administration (2008), and Williamson et al. (2001), driver fatigue is a significant factor in approximately fifteen to twenty per cent of commercial road transport crashes. In Europe it is estimated that one out of two long haul drivers has fallen asleep while driving. One out of five long distance road transport drivers in Australia reported at least one fatigue related incident on their last trip and one out of three drivers reported breaking road rules on at least half of their trips. Many drivers feel that fatigue is a substantial problem for the industry and feel that their companies should ease unreasonably tight schedules and should allow more time for breaks and rests during their trips. According to a survey conducted by McCartt et al. (2008), one out of six truck drivers admitted to have dozed at wheel in the month prior to the survey and revealed that less than one out of two truck drivers reported that delivery schedules are always realistic. Truck drivers who reported that they are frequently given unrealistic delivery schedules are approximately three times as likely to violate the work rules as drivers who rarely or never have to deal with unrealistic delivery schedules.

In their efforts to increase road safety and improve working conditions of truck drivers, governments in Australia, Canada, Europe and the United States have independently adopted new regulations concerning driving and working hours of truck drivers. These regulations impose maximum limits on the amount of driving and working within certain time periods and minimum requirements on the amount and duration of break and rest periods which must be taken by the truck drivers. Compulsory break and rest periods have a significant impact on total travel times which are typically more than twice as long as the pure driving time required in long distance haulage. Ignoring compulsory break and rest periods when generating schedules for truck drivers can lead to unrealistic expectations, large delays, and or violation of driving and working hour regulations. This results in poor working conditions of truck drivers, reduced road safety, and low customer satisfaction.

One of the first research works explicitly considering compulsory break periods within vehicle scheduling is presented by Savelsbergh and Sol (1998) who consider a problem in which lunch breaks and night breaks must be taken within fixed time intervals. Hours of service rules imposed by the U.S. Department of Transportation are first studied by Xu et al. (2003) who present a column generation approach for combined vehicle routing and scheduling. Archetti and Savelsbergh (2009) show that truck driver scheduling problems considering U.S. hours of service regulations can be solved polynomial time. Goel and Kok (2010) present an improved algorithm which can solve truck driver scheduling problems in the Unites States in quadratic time. Recently, several works considering the generation of truck driver schedules complying with European Union regulations have been presented. The first method which is guaranteed to find a truck driver schedule complying with European Union regulations if such a schedule exists is presented by Goel (2010). Goel (2009), Kok et al. (2010), and Prescott-Gagnon et al. (2010) solve combined vehicle routing and truck driver scheduling problems in the European Union by heuristically determining truck driver schedules. To the best of our knowledge Canadian Commercial Vehicle Drivers Hours of Service Regulations have not yet been studied.

In this paper we study and present the Canadian Truck Driver Scheduling Problem, which is the problem of determining whether it is possible to schedule driving and working hours of truck drivers in such a way that a sequence of locations can be visited within given time windows and that Canadian Commercial Vehicle Drivers Hours of Service Regulations are complied with. We evaluate the performance of an exact scheduling algorithm and compare it with two heuristic approaches which require significantly less computation time. Futhermore, we compare the impact on route feasibility of Canadian and United States regulations and analyse the impact of the additional constraints found in Canadian regulations.

The remainder of this paper is organised as follows. Section 2 describes Canadian Commercial Vehicle Drivers Hours of Service Regulations which are discussed in Section 3. In Section 4 we present the notation required in this paper and give a mathematical formulation of the Canadian Truck Driver Scheduling Problem. In Section 5 we present some dominance criteria which help us in effectively tackling the Canadian Truck Driver Scheduling Problem. Section 6 presents a solution framework which is used by the heuristic and exact solution approaches presented in Section 7. In Section 8 we report on computational experiments demonstrating the effectiveness of our methods. Finally, Section 9 gives some concluding remarks.

2 Canadian Commercial Vehicle Drivers Hours of Service Regulations

Canadian regulations concerning driving and working hours of commercial vehicles are described in Transport Canada (2005) and interpreted in Canadian Council of Motor Transport Administrators (2007). In Canada two sets of regulations exists, one of which applies to driving conducted south of latitude 60° N and one to driving north of latitude 60° N. In the remainder of this paper we focus on the subset of regulations applicable for driving south of latitude 60° N because this is the area of major economic concern. The regulation defines *on-duty time* as the period that begins when a driver begins work and ends when the driver stops work and includes any time during which the driver is driving or conducting any other work. *Off-duty time* is defined by any period other than on-duty time. The regulation imposes restrictions on the maximum amount of on-duty time and the minimum amount of off-duty time during a *day*. According to the regulation a *day* means a 24-hour period that begins at some time designated by the motor carrier. For simplicity and w.l.o.g. let us assume in the remainder that this time is midnight.

The regulation demands that a driver must not drive after accumulating 13 hours of driving time, after accumulating 14 hours of on-duty time, or after 16 hours of time have elapsed since the end of the last period of at least 8 consecutive hours of off-duty time. In any of these cases the driver may only commence driving again after taking another period of at least 8 consecutive hours of off-duty time.

The regulation demands that a driver does not drive for more than 13 hours in a day and that a driver accumulates at least 10 hours of off-duty time in a day. At least 2 of these hours must not be part of a period of 8 consecutive hours of off-duty time as required by the provisions described in the previous paragraph. However, if a period of more than 8 consecutive hours of off-duty time is scheduled, the amount exceeding the 8th hour may contribute to these 2 hours. Periods of less than 30 minutes, in which the driver is neither driving nor working, do not count toward the minimum off-duty time requirements given by the regulation, even though they are considered as off-duty time by the definition.

The regulation imposes additional constraints on the amount of on-duty time within a period of 7 days and gives some extra flexibility in scheduling off-duty periods. For the sake of conciseness, however, we will not consider the corresponding provisions of the regulation in the remainder of this paper.

3 Discussion

Canadian Commercial Vehicle Drivers Hours of Service Regulations share some similarities with U.S. hours of service regulations which are studied in Xu et al. (2003), Archetti and Savelsbergh (2009), and Goel and Kok (2010). In Canada a driver may accumulate 13 hours of driving time before taking an off-duty period of 8 consecutive hours, whereas in the United States a driver must take 10 hours of consecutive off-duty time after accumulating 11 hours of driving time. After taking 10 hours of off-duty time, a driver in the United States may commence driving again. Thus, a driver may accumulate a total of 14 hours of driving within a single day. In Canada the maximum amount of driving on a single day is limited to 13 hours. Canadian regulations ensure that on any day at least 10 hours of off-duty time are taken. However, the minimum amount of continuous off-duty time is only 8 hours. In a study on sleeping patterns of truck drivers by Mitler et al. (1997), Canadian truck drivers regularly drove up to 13 hours between rest periods of 8 consecutive hours. The study revealed that on average drivers sleep less than 5 hours per day which is 2 hours less than the average ideal reported by the drivers. More than half of the drivers had at least one six-minute interval of drowsiness while driving within the five-day study. This indicates that the required amount of 8 hours continuous off-duty time may not be sufficient to guarantee that drivers get enough sleep. Undoubtedly, regular sleep deficits may be a contributor to fatigue related road accidents.

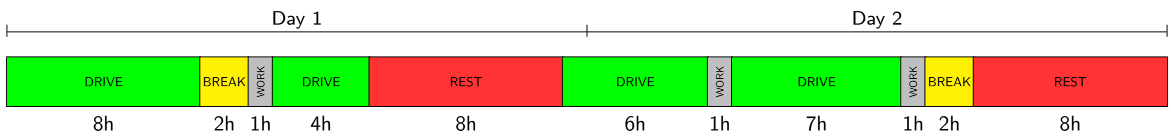

Figure 1: Schedule complying with the regulation

Figure 1 shows a driver schedule complying with Canadian regulations. The driver begins with a driving period and reaches the next customer location at 8.00 AM. Then, the driver takes two hours of off-duty time before beginning to load the vehicle. After one hour of stationary work time, the driver continuous driving towards the next customer location. From 3.00 PM to 11.00 PM the driver takes a rest period. As the driver has not yet reached the maximum allowed amount of driving on day 1, the driver may continue to drive after the rest. After the end of the last on-duty time, the driver must take 10 hours of off-duty time to reach the minimum amount of off-duty time on day 2.

Let us now assume that the schedule in Figure 1 is the planned schedule of a driver. If, for any reason, the driver deviates from the plan and rests until 11.30 PM on day 1, thirty minutes of driving time are pushed from day 1 into day 2 and thirty minutes of off-duty time are pushed from day 2 into day 3. Consequently, the driver does not take the minimum amount of off-duty time required on day 2 and violates against the regulation. We can see that in this example, taking additional off-duty time is actually penalised by the regulation and the driver may be tempted to take a short rest period regardless of his or her actual fitness for duty. This is unique to Canadian regulations and we believe that this is not intended by the legislator. In the remainder of this paper we will thus only consider schedules in which the duration of rest periods can be extended by any amount without violating the regulation.

4 The Truck Driver Scheduling Problem

In this section we present a formal model of the Canadian Truck Driver Scheduling Problem which is the problem of visiting a sequence of λ locations within given time windows in such a way that driving and working activities of truck drivers comply with Canadian Commercial Vehicle Drivers Hours of Service Regulations. Let us first present the notation used throughout the paper and describe under which conditions a truck driver schedule complies with the regulation. The parameters imposed by the regulation are summarised in Table 1.

In order to represent truck driver schedules, let us denote with `DRIVE` any on-duty time during which the driver is driving, with `WORK` any on-duty time during which the driver is not driving. Furthermore, let us denote with `REST` any period of 8 hours or more of off-duty time which resets the accumulated amount of driving, the accumulated amount of on-duty time, and the time elapsed since the last period of 8 consecutive hours of off-duty time. Let us denote with `BREAK` any period contributing to the off-duty requirements which is not interpreted as rest period. Note that it is possible to take a `BREAK` period immediately before or after a `REST` period. If a `BREAK` period is taken immediately before or after a `REST` period, the duration of the `BREAK` period may be shorter than

Notation	Value	Description
t^{rest}	8 h	The minimum duration of a rest period
t^{drive}	13 h	The maximum accumulated driving time between two consecutive rest periods and the maximum accumulated driving time on a day
$t^{\text{on-duty}}$	14 h	The maximum accumulated on-duty time until a driver may drive between two consecutive rest periods
t^{elapsed}	16 h	The maximum time since the end of the off-duty period commencing with the last rest period
t^{day}	24 h	The duration of a day
$t^{\text{off-duty}}$	10 h	The minimum amount of off-duty time on a day
t^{break}	2 h	The minimum amount of off-duty time on a day which is not part of a rest period
t^{length}	$\frac{1}{2}$ h	The minimum length of an off-duty period period to be counted

Table 1: Parameters imposed by the regulation

$\frac{1}{2}$ hour, as both periods can be interpreted as one continuous block of off-duty time. For the same reason, the time elapsed since the end of the last period of 8 consecutive hours of off-duty time does not increase if a `BREAK` period is taken immediately after a `REST` period. Let us denote with `IDLE` any other off-duty time of less than $\frac{1}{2}$ hour duration which does not contribute to the off-duty requirements of the regulation. Note, that idle periods are only required between on-duty periods, because the duration of a preceding or suceeding off-duty period could be increased otherwise.

A truck driver schedule can be specified by a sequence of activities to be performed by the driver. Let $\mathcal{A} := \{a = (a^{\text{type}}, a^{\text{length}}) \mid a^{\text{type}} \in \{\texttt{DRIVE}, \texttt{WORK}, \texttt{REST}, \texttt{BREAK}, \texttt{IDLE}\}, a^{\text{length}} > 0\}$ denote the set of driver activities to be scheduled. Let « . » be an operator which 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. During concatenation the operator merges consecutive driver activities of the same type. That is, for a given schedule $s := a_1.a_2. \ldots .a_k$ and an activity a with $a_k^{\text{type}} = a^{\text{type}}$ we have $s.a = a_1. \ldots .a_{k-1}.(a_k^{\text{type}}, a_k^{\text{length}} + a^{\text{length}})$. For a given schedule $s := a_1.a_2. \ldots .a_k$ and $1 \leq i \leq k$ let $s_{1,i} := a_1.a_2. \ldots .a_i$ denote the partial schedule composed of activities a_1 to a_i. We assume that at the beginning of the planning horizon, the driver returns from a rest period which is long enough such that previous driving and working activities do not have any influence on the driving and working hours within the planning horizon. We will thus only consider schedules $s := a_1.a_2. \ \ldots \ .a_k$ which begin with a rest period, i.e. $a_1^{\text{type}} = \texttt{REST}$.

For a given schedule $s := a_1.a_2. \ \ldots \ .a_k$ with $a_1^{\text{type}} = \texttt{REST}$ let l_s^{end} denote the completion time of the schedule, let l_s^{drive} denote the accumulated driving time since the end of the last rest period, and let $l_s^{\text{on-duty}}$ denote the accumulated on-duty time since the end of the last rest period, and let l_s^{idle} denote the accumulated idle time since the end of the last rest period. These values can be recursively computed during schedule generation by setting $l_{s_{1,1}}^{\text{end}} := a_1^{\text{length}}$, $l_{s_{1,1}}^{\text{drive}} := 0$, $l_{s_{1,1}}^{\text{on-duty}} := 0$, $l_{s_{1,1}}^{\text{idle}} := 0$, and

$$l_{s.a}^{\text{end}} := l_s^{\text{end}} + a^{\text{length}},$$

$$l_{s.a}^{\text{drive}} \quad := \begin{cases} 0 & \text{if } a^{\text{type}} = \texttt{REST} \\ l_s^{\text{drive}} + a^{\text{length}} & \text{else if } a^{\text{type}} = \texttt{DRIVE} \\ l_s^{\text{drive}} & \text{else} \end{cases}$$

$$l_{s.a}^{\text{on-duty}} \quad := \begin{cases} 0 & \text{if } a^{\text{type}} = \texttt{REST} \\ l_s^{\text{on-duty}} + a^{\text{length}} & \text{else if } a^{\text{type}} \in \{\texttt{DRIVE}, \texttt{WORK}\} \\ l_s^{\text{on-duty}} & \text{else} \end{cases}$$

$$l_{s.a}^{\text{idle}} \quad := \begin{cases} 0 & \text{if } a^{\text{type}} = \texttt{REST} \\ l_s^{\text{idle}} + a^{\text{length}} & \text{else if } a^{\text{type}} = \texttt{IDLE} \\ l_s^{\text{idle}} & \text{else} \end{cases}$$

The restrictions that a driver must not drive after accumulating 13 hours of driving time and that a driver must not drive after accumulating 14 hours of on-duty time are satisfied if

$$l_{s_{1,i}}^{\text{drive}} \leq t^{\text{drive}} \text{ for any } 1 < i \leq k \text{ with } a_i^{\text{type}} = \texttt{DRIVE}$$

and

$$l_{s_{1,i}}^{\text{on-duty}} \leq t^{\text{on-duty}} \text{ for any } 1 < i \leq k \text{ with } a_i^{\text{type}} = \texttt{DRIVE}$$

Let l_s^{elapsed} denote the amount of time which has elapsed since the end of the off-duty period commencing with the last rest period. This value can be recursively computed during schedule generation by setting $l_{s_{1,1}}^{\text{elapsed}} := 0$ and

$$l_{s.a}^{\text{elapsed}} := \begin{cases} 0 & \text{if } a^{\text{type}} = \texttt{REST} \\ 0 & \text{if } a^{\text{type}} = \texttt{BREAK} \text{ and } l_s^{\text{elapsed}} = 0 \\ l_s^{\text{elapsed}} + a^{\text{length}} & \text{else} \end{cases}$$

Note, that we can append any amount of break to a rest period without increasing l_s^{elapsed}. Furthermore, note that for notational reasons, we do not reset l_s^{elapsed}, if a break period of 8 hours or more is scheduled. In order to reset l_s^{elapsed}, a rest period can be scheduled instead of a break period of 8 hours or more. A schedule satisfies the restriction that no driving is conducted after 16 hours of time have elapsed since the end of the off-duty period commencing with the last rest period if

$$l_{s_{1,i}}^{\text{elapsed}} \leq t^{\text{elapsed}} \text{ for any } 1 < i \leq k \text{ with } a_i^{\text{type}} = \texttt{DRIVE}$$

We must assure that rest activities have the minimum duration required by the regulation, i.e.

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

Let l_s^{length} denote the minimum duration a break period which is appended to schedule s must have to be counted as off-duty time. This value can be recursively computed during schedule generation by setting $l_{s_{1,1}}^{\text{length}} := 0$ and

$$l_{s.a}^{\text{length}} := s \begin{cases} 0 & \text{if } a^{\text{type}} \in \{\texttt{REST}, \texttt{BREAK}\} \\ t^{\text{length}} & \text{else} \end{cases}$$

Each break period that contributes to the 2 hours of off-duty time during a day must have a duration of at least 30 minutes or must precede or succeed a rest period, i.e. each schedule must satisfy

$$a_i^{\text{length}} \geq l_{s_{1,i-1}}^{\text{length}} \text{ for any } 1 < i < k \text{ with } a_i^{\text{type}} = \texttt{BREAK}, a_{i+1}^{\text{type}} \neq \texttt{REST}$$

Let us assume that the planning horizon begins at day 1 and ends at the day denoted by d^{max}. Let $l_s^{\text{drive}|d}$ denote the accumulated driving time on day d. This value can be recursively computed for all $d \in \{1, 2, \ldots, d^{\text{max}}\}$ by setting $l_{s_{1,1}}^{\text{drive}|d} := 0$ and

$$l_{s.a}^{\text{drive}|d} := \begin{cases} l_s^{\text{drive}|d} & \text{if } a^{\text{type}} \neq \texttt{DRIVE} \text{ or } l_{s.a}^{\text{end}} \leq (d-1) \cdot t^{\text{day}} \text{ or } l_s^{\text{end}} \geq d \cdot t^{\text{day}} \\ l_s^{\text{drive}|d} + \min\{d \cdot t^{\text{day}}, l_{s.a}^{\text{end}}\} - \max\{(d-1) \cdot t^{\text{day}}, l_s^{\text{end}}\} & \text{else} \end{cases}$$

A schedule satisfies the restriction that no driving is conducted after accumulating 13 hours of driving during a day if

$$l_s^{\text{drive}|d} \leq t^{\text{drive}} \text{ for any } d \in \{1, 2, \ldots, d^{\max}\}$$

Let $l_s^{\text{off-duty}|d}$ denote the accumulated off-duty time on day d. This value can be recursively computed for all $d \in \{1, 2, \ldots, d^{\max}\}$ by setting $l_{s_{1,1}}^{\text{off-duty}|d} := 0$ and

$$l_{s.a}^{\text{off-duty}|d} := \begin{cases} l_s^{\text{off-duty}|d} & \text{if } a^{\text{type}} \notin \{\texttt{BREAK}, \texttt{REST}\} \text{ or } l_{s.a}^{\text{end}} \leq (d-1) \cdot t^{\text{day}} \text{ or } l_s^{\text{end}} \geq d \cdot t^{\text{day}} \\ l_s^{\text{off-duty}|d} + \min\{d \cdot t^{\text{day}}, l_{s.a}^{\text{end}}\} - \max\{(d-1) \cdot t^{\text{day}}, l_s^{\text{end}}\} & \text{else} \end{cases}$$

A schedule satisfies the restriction that at least 10 hours of off-duty time are scheduled during a day if

$$l_s^{\text{off-duty}|d} \geq t^{\text{off-duty}} \text{ for any } d \in \{1, 2, \ldots, d^{\max}\}$$

Let $l_s^{\text{break}|d}$ denote the accumulated off-duty time on day d which is not part of a rest period. This value can be recursively computed for all $d \in \{1, 2, \ldots, d^{\max}\}$ by setting $l_{s_{1,1}}^{\text{break}|d} := 0$ and

$$l_{s.a}^{\text{break}|d} := \begin{cases} l_s^{\text{break}|d} & \text{if } a^{\text{type}} \neq \texttt{BREAK} \text{ or } l_{s.a}^{\text{end}} \leq (d-1) \cdot t^{\text{day}} \text{ or } l_s^{\text{end}} \geq d \cdot t^{\text{day}} \\ l_s^{\text{break}|d} + \min\{d \cdot t^{\text{day}}, l_{s.a}^{\text{end}}\} - \max\{(d-1) \cdot t^{\text{day}}, l_s^{\text{end}}\} & \text{else} \end{cases}$$

A schedule satisfies the restriction that at least 2 hours of off-duty time are scheduled during a day which are not part of a rest period if

$$l_s^{\text{break}|d} \geq t^{\text{break}} \text{ for any } d \in \{1, 2, \ldots, d^{\max}\}$$

As we only consider schedules in which the duration of rest periods can be extended without violating the regulation, we demand that

$$l_{s_{1,i}}^{\text{on-duty}} + l_{s_{1,i}}^{\text{idle}} \leq t^{\text{day}} - t^{\text{off-duty}} \text{ for any } 1 < i \leq k$$

Under this condition rest periods can be extended by any value while leaving enough time for the required off-duty time on the subsequent day.

Let us now give formal model of the Canadian Truck Driver Scheduling Problem. Let us consider a sequence of locations denoted by $n_1, n_2, \ldots, n_\lambda$ which shall be visited by a truck driver. At each location n_μ some stationary work of duration w_μ shall be conducted. This work must be a continuous period which shall begin within a time window denoted by $[t_\mu^{\min}, t_\mu^{\max}]$. We assume that n_1 corresponds to the driver's current location and that the driver completes her or his work week after finishing work at location n_λ. The work to be conducted at locations n_1 and n_λ can include loading and unloading activities as well as time for getting ready or cleaning the vehicle. The (positive) driving time required for moving from node n_μ to node $n_{\mu+1}$ shall be denoted by $\delta_{\mu,\mu+1}$. Let us assume that all values representing driving times, working times, and time windows are a multiple of 15 minutes.

For a given sequence of locations $n_1, n_2, \ldots, n_\lambda$ and a schedule $s = a_1.a_2.\ \ldots\ .a_k$ with $a_1^{\text{type}} = \texttt{REST}$, let us denote with $i(\mu)$ the index corresponding to the μth stationary work period, i.e. $a_{i(\mu)}$ corresponds to the work performed at location n_μ. The Canadian Truck Driver Scheduling Problem (CAN-TDSP) is the problem of determining whether a schedule $s := a_1.a_2.\ \ldots\ .a_k$ with $a_1^{\text{type}} = \texttt{REST}$ exists which satisfies

$$\sum_{\substack{1\le j\le k\\ a_j^{\text{type}}=\texttt{WORK}}} 1 = \lambda \tag{1}$$

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

$$t_\mu^{\text{min}} \le l_{s_{1,i(\mu)-1}}^{\text{end}} \le t_\mu^{\text{max}} \text{ for each } \mu \in \{1,2,\ldots,\lambda\} \tag{3}$$

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

$$l_{s_{1,i}}^{\text{drive}} \le t^{\text{drive}} \text{ for any } 1 < i \le k \tag{5}$$

$$l_{s_{1,i}}^{\text{on-duty}} \le t^{\text{on-duty}} \text{ for any } 1 < i \le k \text{ with } a_i^{\text{type}} = \texttt{DRIVE} \tag{6}$$

$$l_{s_{1,i}}^{\text{elapsed}} \le t^{\text{elapsed}} \text{ for any } 1 < i \le k \text{ with } a_i^{\text{type}} = \texttt{DRIVE} \tag{7}$$

$$a_i^{\text{length}} \ge t^{\text{rest}} \text{ for any } 1 < i \le k \text{ with } a_i^{\text{type}} = \texttt{REST} \tag{8}$$

$$a_i^{\text{length}} \ge l_{s_{1,i-1}}^{\text{length}} \text{ for any } 1 < i < k \text{ with } a_i^{\text{type}} = \texttt{BREAK}, a_{i+1}^{\text{type}} \ne \texttt{REST} \tag{9}$$

$$l_s^{\text{drive}|d} \le t^{\text{drive}} \text{ for any } d \in \{1,2,\ldots,d^{\text{max}}\} \tag{10}$$

$$l_s^{\text{off-duty}|d} \ge t^{\text{off-duty}} \text{ for any } d \in \{1,2,\ldots,d^{\text{max}}\} \tag{11}$$

$$l_s^{\text{break}|d} \ge t^{\text{break}} \text{ for any } d \in \{1,2,\ldots,d^{\text{max}}\} \tag{12}$$

$$l_{s_{1,i}}^{\text{on-duty}} + l_{s_{1,i}}^{\text{idle}} \le t^{\text{day}} - t^{\text{off-duty}} \text{ for any } 1 < i \le k \tag{13}$$

Condition (1) demands that the number of work activities in the schedule is λ. 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 corresponding 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) to (12) are the constraints imposed by the regulation. Condition (13) guarantees that rest periods can be extended without violating the regulation.

In the remainder of this paper, we will say that a schedule $s := a_1.a_2. \ \ldots \ .a_k$ with $a_1^{\text{type}} = \texttt{REST}$ is *feasible* if it satisfies conditions (1) to (13). The solution approaches presented in this paper will generate partial schedules which are iteratively modified in order to find a feasible schedule for the CAN-TDSP. All partial schedules generated in the solution approaches will satisfy conditions (1) to (4) given that λ is replaced with λ', where λ' is the number of work activities in the partial schedule. Furthermore, the accumulated driving time since the last work activity in the partial schedule will not exceed $\delta_{\lambda',\lambda'+1}$. Let us say that the day $d \in \{1, 2, \ldots, d^{\max}\}$ is the *current day* of a partial schedule s if $(d-1) \cdot t^{\text{day}} \leq l_s^{\text{end}} < d \cdot t^{\text{day}}$. A partial schedule complies with the regulations if conditions (5) to (9) are satisfied, conditions (10) to (12) are satisfied for all days prior to the current day, and for the current day condition (10) is satisfied and conditions (11) and (12) can be satisfied by appending a break period of sufficient duration. In the remainder of this paper we will say that a partial schedule is feasible if above conditions are satisfied.

5 Dominance

In general there are too many different alternative feasible partial schedules to be considered in order to fully enumerate the search space using the « . » operator. In this section we present dominance criteria that help us reducing the number of partial schedules that need to be considered when solving the CAN-TDSP. For this, let us first define some additional operators which allow for inserting break activities into the schedule. These operators allow us to only append rest and break periods of minimal duration to a schedule because we can increase the off-duty time later on.

Let «$\overset{\texttt{R}}{\leftarrow}$» be an operator which inserts a break activity of given duration after the last activity of type REST and let «$\overset{\texttt{B}}{\leftarrow}$» be an operator which inserts a break activity of given duration after the last activity of type BREAK. If no break is scheduled after the last rest period, «$\overset{\texttt{B}}{\leftarrow}$» inserts the break after the last activity of type REST. That is, if i is the index of the last rest activity and j is the index of the last break or rest activity in schedule $s := a_1.a_2. \ \ldots \ .a_k$ and $\Delta > 0$, then we have $s \overset{\texttt{R}}{\leftarrow} \Delta = a_1.a_2. \ \ldots \ a_i.(\texttt{BREAK}, \Delta).a_{i+1}. \ \ldots \ .a_k.$ and $s \overset{\texttt{B}}{\leftarrow} \Delta = a_1.a_2. \ \ldots \ a_j.(\texttt{BREAK}, \Delta).a_{j+1}. \ \ldots \ .a_k.$

Inserting a break activity after the last rest or break activity can result in a violation of time window constraints. Furthermore, inserting a break activity after the last break could push the end of some driving activity to a value exceeding the maximum time that may elapse after the end of the last off-duty period commencing with the last rest activity. For any schedule $s := a_1.a_2. \ \ldots \ .a_k$, let us denote with $\mu(s)$ the index of the next work location to be visited. Let $l_s^{\text{push}|\texttt{R}}$ and $l_s^{\text{push}|\texttt{B}}$ denote the maximum amount by which activities succeeding the break activity which is inserted by «$\overset{\texttt{R}}{\leftarrow}$» or «$\overset{\texttt{B}}{\leftarrow}$» may be pushed into the future without violating time window constraints or condition (7). These values can be recursively computed during schedule generation by setting $l_{s_{1,1}}^{\text{push}|\texttt{R}} := \infty$,

$$l_{s.a}^{\text{push}|\texttt{R}} := \begin{cases} \infty & \text{if } a^{\text{type}} = \texttt{REST} \\ \min\{l_s^{\text{push}|\texttt{R}}, t_{\mu(s)}^{\max} - l_s^{\text{end}}\} & \text{if } a^{\text{type}} = \texttt{WORK} \\ l_s^{\text{push}|\texttt{R}} & \text{else} \end{cases}$$

and $l_{s_{1,1}}^{\text{push}|\texttt{B}} := \infty$,

$$l_{s.a}^{\text{push}|\texttt{B}} := \begin{cases} \infty & \text{if } a^{\text{type}} \in \{\texttt{REST}, \texttt{BREAK}\} \\ \min\{l_s^{\text{push}|\texttt{B}}, t_{\mu(s)}^{\max} - l_s^{\text{end}}\} & \text{if } a^{\text{type}} = \texttt{WORK} \\ \min\{l_s^{\text{push}|\texttt{B}}, t^{\text{elapsed}} - l_{s.a}^{\text{elapsed}}\} & \text{if } a^{\text{type}} = \texttt{DRIVE} \\ l_s^{\text{push}|\texttt{B}} & \text{else} \end{cases}$$

A partial schedule obtained by applying $s \overset{\texttt{R}}{\leftarrow} \Delta$ or $s \overset{\texttt{B}}{\leftarrow} \Delta$ is feasible if and only if $\Delta \leq l_s^{\text{push}|\texttt{R}}$ or $\Delta \leq l_s^{\text{push}|\texttt{B}}$. If a feasible schedule exists for an instance of the constrained CAN-TDSP, we can efficiently determine this schedule by applying a sequence of operators moves to the initial schedule $s := (\texttt{REST}, t^{\text{rest}})$. Here, each operator move either appends an activity to the schedule, inserts a break period after the last rest period, or inserts a break period after the last break period.

Let us now consider two feasible partial schedules s' and s''. If for any feasible schedule that can be generated by applying a sequence of operator moves to schedule s'', we can generate a feasible schedule by applying a sequence of operator moves to schedule s', then s' *dominates* s''.

Lemma 1 Let s' and s'' be feasible partial schedules which have the same amount of accumulated on-duty time. Furthermore, let us denote with d the current day of schedule s'. If $l_{s'}^{\text{end}} = l_{s''}^{\text{end}}$, $l_{s'}^{\text{drive}} \leq l_{s''}^{\text{drive}}$, $l_{s'}^{\text{on-duty}} \leq l_{s''}^{\text{on-duty}}$, $l_{s'}^{\text{idle}} \leq l_{s''}^{\text{idle}}$, $l_{s'}^{\text{elapsed}} \leq l_{s''}^{\text{elapsed}}$, $l_{s'}^{\text{drive}|d} \leq l_{s''}^{\text{drive}|d}$, $l_{s'}^{\text{break}|d} \geq l_{s''}^{\text{break}|d}$, $l_{s'}^{\text{off-duty}|d} \geq l_{s''}^{\text{off-duty}|d}$, $l_{s'}^{\text{push}|\mathtt{R}} \geq l_{s''}^{\text{push}|\mathtt{R}}$, $l_{s'}^{\text{push}|\mathtt{B}} \geq l_{s''}^{\text{push}|\mathtt{B}}$, and if for all $\Delta \in [0, l_{s'}^{\text{elapsed}}]$ the accumulated amount of break time in the last Δ minutes of schedule s' is at least as high as the corresponding value for schedule s'' and the accumulated amount of driving time in the last Δ minutes of schedule s' is not higher than the corresponding value for schedule s'', then s' dominates s''.

Lemma 2 Let s' and s'' be feasible partial schedules which have the same amount of accumulated on-duty time. Furthermore, let us denote with d the current day of schedule s'. If $l_{s'}^{\text{end}} + \max\{0, t^{\text{break}} - l_{s'}^{\text{break}|d}\} + t^{\text{off-duty}} \leq l_{s''}^{\text{end}}$, and if $l_{s''}^{\text{end}} \geq d \cdot t^{\text{day}}$ or $l_{s'}^{\text{drive}|d} \leq l_{s''}^{\text{drive}|d}$, then s' dominates schedule s''.

The proofs of these lemmata can be found in the Appendix. Note, that we can use Lemma 1 to derive other dominance criteria. For example, under certain conditions we can show dominance of a schedule s' over a schedule s'' if $l_{s'}^{\text{end}} < l_{s''}^{\text{end}}$. If $l_{s'}^{\text{end}} + \min\{l_{s''}^{\text{end}} - l_{s'}^{\text{end}}, l_{s''}^{\text{elapsed}} - l_{s'}^{\text{elapsed}}\} \leq l_{s''}^{\text{end}}$ and $l_{s'}^{\text{push}|\mathtt{R}} \geq l_{s''}^{\text{end}} - l_{s'}^{\text{end}} - \min\{l_{s''}^{\text{end}} - l_{s'}^{\text{end}}, l_{s''}^{\text{elapsed}} - l_{s'}^{\text{elapsed}}\}$, we can set $\hat{s}' := s'.(\mathtt{BREAK}, \min\{l_{s''}^{\text{end}} - l_{s'}^{\text{end}}, l_{s''}^{\text{elapsed}} - l_{s'}^{\text{elapsed}}\})$ and $\bar{s}' := \hat{s}' \overset{\mathtt{R}}{\leftarrow} (l_{s''}^{\text{end}} - l_{\hat{s}'}^{\text{end}})$. If the conditions of Lemma 1 hold for $\bar{s}'$ and s'', then s' dominates s''.

The dominance criteria can not only be used to discard some of the partial schedules which are generated throughout the solution process, but they can also be used to derive some guidelines allowing us to develop efficient solution approaches which only generate the most promising partial schedules. First, if it is possible to drive or work no off-duty period should be scheduled. If necessary, the off-duty period can be appended after the driving or working period or a break can later be inserted after the last rest or break period in the schedule. Second, a rest period should only be scheduled if $l_s^{\text{elapsed}} > 0$ and every rest period should have a duration of exactly t^{rest}, because otherwise we can schedule break time instead. Third, the amount of break time scheduled should be as small as possible, because we can append or insert additional break time later. The solution framework presented in the next section uses these guidelines to minimise the number of schedules which are generated.

6 Solution Framework

In this section we present a solution framework for solving the Canadian Truck Driver Scheduling Problem which takes a set of feasible schedules for a partial tour $n_1, n_2, \ldots, n_\mu$ and extends each schedule to construct feasible schedules for tour $n_1, n_2, \ldots, n_\mu, n_{\mu+1}$. Let $\mathcal{S}_\mu$ denote the set of feasible schedules found for the partial tour $n_1, n_2, \ldots, n_\mu$. In the beginning of our solution approach we set

$$\mathcal{S}_1 := \begin{cases} \{(\texttt{REST}, t^{\text{rest}}).(\texttt{WORK}, w_1)\} & \text{if } t_1^{\min} \leq t^{\text{rest}} \\ \{(\texttt{REST}, t^{\text{rest}}).(\texttt{BREAK}, t_1^{\min} - t^{\text{rest}}).(\texttt{WORK}, w_1)\} & \text{if } t_1^{\min} > t^{\text{rest}} \end{cases}$$

and

$$\mathcal{S}_\mu := \emptyset \text{ for all } 1 < \mu \leq \lambda.$$

We set $\mu := 1$ and determine $\mathcal{S}_{\mu+1}$. This process is repeated with $\mu := \mu + 1$ until the CAN-TDSP for tour $n_1, n_2, \ldots, n_\lambda$ is solved.

The scheduling framework is composed of two parts: the first part schedules all activities on the trip from node n_μ to node $n_{\mu+1}$; the second part schedules the stationary activities after the (physical) arrival at location $n_{\mu+1}$.

The method for scheduling activities on a trip from node n_μ to node $n_{\mu+1}$ is illustrated in Figure 2. In the method we denote with δ_s the remaining driving time required to reach the next location $n_{\mu+1}$.

The method starts by initialising the set of partial schedules $\mathcal{S}$ which is set to $\mathcal{S}_\mu$ and the initially empty set $\mathcal{S}'_{\mu+1}$ of schedules in which no further driving is required to reach location $n_{\mu+1}$. Then, it chooses a partial schedule $s \in \mathcal{S}$ and removes it from $\mathcal{S}$. The method determines the amount of break still required on the current day d by setting $\Delta^{\text{break}} := \max\{0, t^{\text{break}} - l_s^{\text{break}|d}\}$, the amount of off-duty time still required on the current day d by setting $\Delta^{\text{off-duty}} := \max\{0, t^{\text{off-duty}} - l_s^{\text{off-duty}|d}\}$, and the maximum amount of driving that can be appended to s by setting

$$\Delta^{\text{drive}} := \min\{\delta_s, t^{\text{drive}} - l_s^{\text{drive}}, t^{\text{on-duty}} - l_s^{\text{on-duty}}, t^{\text{elapsed}} - l_s^{\text{elapsed}}, t^{\text{day}} - t^{\text{off-duty}} - l_s^{\text{on-duty}} - l_s^{\text{idle}}\}.$$

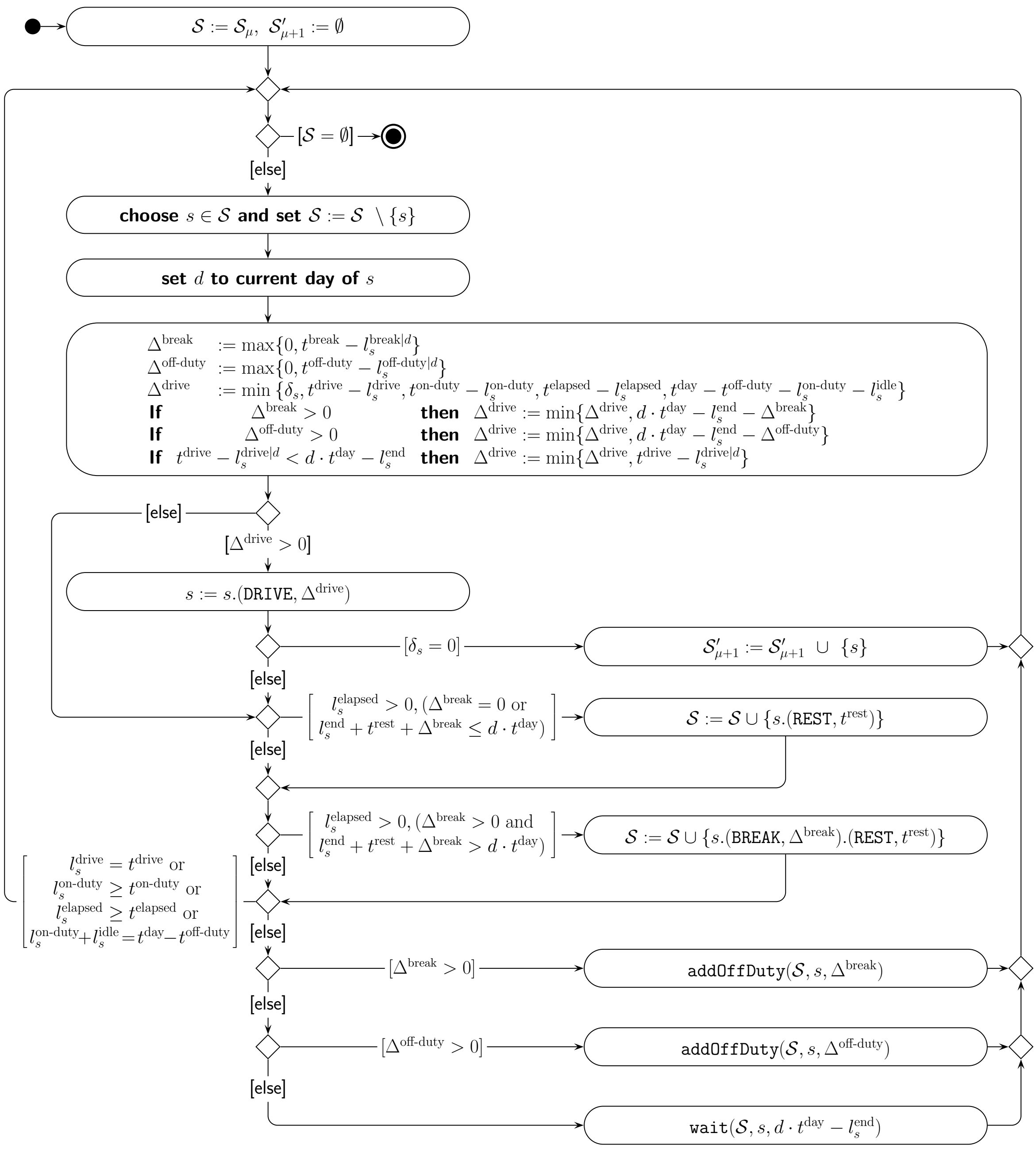

Figure 2: Method for scheduling activities on a trip from n_μ to $n_{\mu+1}$

If $\Delta^{\text{break}} > 0$ and/or $\Delta^{\text{off-duty}} > 0$, some additional off-duty time is required on the current day and the method sets $\Delta^{\text{drive}} := \min\{\Delta^{\text{drive}}, d \cdot t^{\text{day}} - l_s^{\text{end}} - \Delta^{\text{break}}\}$ and/or $\Delta^{\text{drive}} := \min\{\Delta^{\text{drive}}, d \cdot t^{\text{day}} - l_s^{\text{end}} - \Delta^{\text{off-duty}}\}$ to guarantee that enough time for taking the required amount of off-duty remains. If $t^{\text{drive}} - l_s^{\text{drive}|d} < d \cdot t^{\text{day}} - l_s^{\text{end}}$, then the method sets $\Delta^{\text{drive}} := \min\{\Delta^{\text{drive}}, t^{\text{drive}} - l_s^{\text{drive}|d}\}$ to guarantee that the accumulated driving time on the current day will not exceed t^{drive}.

If $\Delta^{\text{drive}} > 0$, the method appends a driving period of duration Δ^{drive} to the schedule. Any other schedules that could be generated by applying some other operator move would be dominated by this schedule generated. If $\delta_s = 0$ after scheduling the driving activity, the next location is reached and the schedule s is included to the set $\mathcal{S}'_{\mu+1}$. If $\Delta^{\text{drive}} \leq 0$ or $\delta_s > 0$ after scheduling the driving activity, some off-duty time must be added to schedule before another driving activity may be scheduled.

If $l_s^{\text{elapsed}} > 0$, some on-duty time has been scheduled since the last rest period. Thus, it may be beneficial to reinitialise l_s^{drive}, $l_s^{\text{on-duty}}$, l_s^{idle}, and l_s^{elapsed} by appending a rest period to the schedule. This, however, can only be made if $\Delta^{\text{break}} = 0$ or $l_s^{\text{end}} + t^{\text{rest}} + \Delta^{\text{break}} \leq d \cdot t^{\text{day}}$, because otherwise the accumulated amount of break time during day d would not be achieved. If $\Delta^{\text{break}} > 0$ and $l_s^{\text{end}} + t^{\text{rest}} + \Delta^{\text{break}} > d \cdot t^{\text{day}}$, we must schedule the required amount of break before we can append a rest period to the schedule. The scheduling method generates a new schedule by appending a rest period which, if necessary, is preceded by a break period of appropriate length and adds the schedule to the set $\mathcal{S}$.

If $l_s^{\text{drive}} = t^{\text{drive}}$, $l_s^{\text{on-duty}} \geq t^{\text{on-duty}}$, $l_s^{\text{elapsed}} \geq t^{\text{elapsed}}$, or $l_s^{\text{on-duty}} + l_s^{\text{idle}} = t^{\text{day}} - t^{\text{off-duty}}$ a rest period is required before further driving can be conducted. As there is no better way to generate a schedule by adding a rest period as described above, the method continues with the next loop.

Otherwise, we have $\Delta^{\text{break}} = d \cdot t^{\text{day}} - l_s^{\text{end}}$, $\Delta^{\text{off-duty}} = d \cdot t^{\text{day}} - l_s^{\text{end}}$, or $l_s^{\text{drive}|d} = t^{\text{drive}}$. We can modify the schedule in such a way that these constraints are no longer binding without scheduling a rest period. If $\Delta^{\text{break}} > 0$ or $\Delta^{\text{off-duty}} > 0$ the method determines new schedules by increasing the amount of off-duty time on day d. Possible implementations of the method `addOffDuty`$(\cdot, \cdot, \cdot)$ which is used to increase the amount of off-duty time by Δ^{break} or $\Delta^{\text{off-duty}}$ are described in the next section. After increasing the accumulated amount of off-duty time the method continuous with the next loop.

If $\Delta^{\text{break}} = 0$ and $\Delta^{\text{off-duty}} = 0$ we have $l_s^{\text{drive}|d} = t^{\text{drive}}$. Thus, no further driving on the current day d is allowed. The methods generates new schedules by increasing the completion time of the schedule to at least $d \cdot t^{\text{day}}$ using the method $\texttt{wait}(\cdot,\cdot,\cdot)$. Possible implementations of this method are also described in the next section. After that, the method continuous with the next loop.

The second part of the scheduling framework, i.e. the method for scheduling stationary activities at node $n_{\mu+1}$, is illustrated in Figure 3. The method starts by initialising the set of partial schedules $\mathcal{S}$ which is set to $\mathcal{S}'_{\mu+1}$. Then, it removes all schedules from $\mathcal{S}$ which have a completion time exceeding the time window of $n_{\mu+1}$. If $\mathcal{S}$ is empty after removing these schedules, the method prematurely terminates because no feasible schedule can be found. Otherwise, it chooses a partial schedule $s \in \mathcal{S}$ and removes it from $\mathcal{S}$. The method determines the amount of break still required on the current day d by setting $\Delta^{\text{break}} := \max\{0, t^{\text{break}} - l_s^{\text{break}|d}\}$, and the amount of off-duty time still required on the current day d by setting $\Delta^{\text{off-duty}} := \max\{0, t^{\text{off-duty}} - l_s^{\text{off-duty}|d}\}$. If $l_s^{\text{end}} \geq t_{\mu+1}^{\text{min}}$, $l_s^{\text{on-duty}} + l_s^{\text{idle}} + w_{\mu+1} \leq t^{\text{day}} - t^{\text{off-duty}}$ and if $\Delta^{\text{break}} = 0$ or $l_s^{\text{end}} + w_{\mu+1} + \Delta^{\text{break}} \leq d \cdot t^{\text{day}}$ and if $\Delta^{\text{off-duty}} = 0$ or $l_s^{\text{end}} + w_{\mu+1} + \Delta^{\text{off-duty}} \leq d \cdot t^{\text{day}}$, the method adds the schedule $s.(\texttt{WORK}, w_{\mu+1})$ to the set $\mathcal{S}_{\mu+1}$ and continues with the next loop as all other schedules that could be generated by applying different operator moves to s would be dominated by this schedule.

Otherwise, some off-duty time is required before the work activity can be appended to the schedule. The off-duty time required may be a rest period, a break period, an idle period, or a combination of these periods.

If $l_s^{\text{elapsed}} = 0$, no on-duty time has been scheduled since the last rest period and there is no reason to add another rest period to the schedule. Otherwise, we may append a rest period to the schedule if $\Delta^{\text{break}} = 0$ or $l_s^{\text{end}} + t^{\text{rest}} + \Delta^{\text{break}} \leq d \cdot t^{\text{day}}$. If $l_s^{\text{elapsed}} > 0$, $\Delta^{\text{break}} > 0$, and $l_s^{\text{end}} + t^{\text{rest}} + \Delta^{\text{break}} > d \cdot t^{\text{day}}$, we must schedule the required amount of break before we may append a rest period to the schedule. The scheduling method generates a new schedule by appending a rest period which, if necessary, is preceded by a break period of appropriate length and adds the schedule to the set $\mathcal{S}$.

If $l_s^{\text{on-duty}} + l_s^{\text{idle}} + w_{\mu+1} > t^{\text{day}} - t^{\text{off-duty}}$, or if $l_s^{\text{elapsed}} > 0$ and $l_s^{\text{end}} + \Delta^{\text{break}} + t^{\text{off-duty}} \leq t_{\mu+1}^{\text{min}}$, no better way to increase the completion time of the schedule exists and the method continues with the next loop.

If $\Delta^{\text{break}} > 0$ and either $t_{\mu+1}^{\text{min}} \geq d \cdot t^{\text{day}}$ or $l_s^{\text{end}} + w_{\mu+1} + \Delta^{\text{break}} > d \cdot t^{\text{day}}$ we must add Δ^{break} of break time to the schedule before the work activity can be scheduled. If $\Delta^{\text{off-duty}} > 0$ and either $t_{\mu+1}^{\text{min}} \geq d \cdot t^{\text{day}}$ or $l_s^{\text{end}} + w_{\mu+1} + \Delta^{\text{off-duty}} > d \cdot t^{\text{day}}$ we must add $\Delta^{\text{off-duty}}$ of off-duty time to

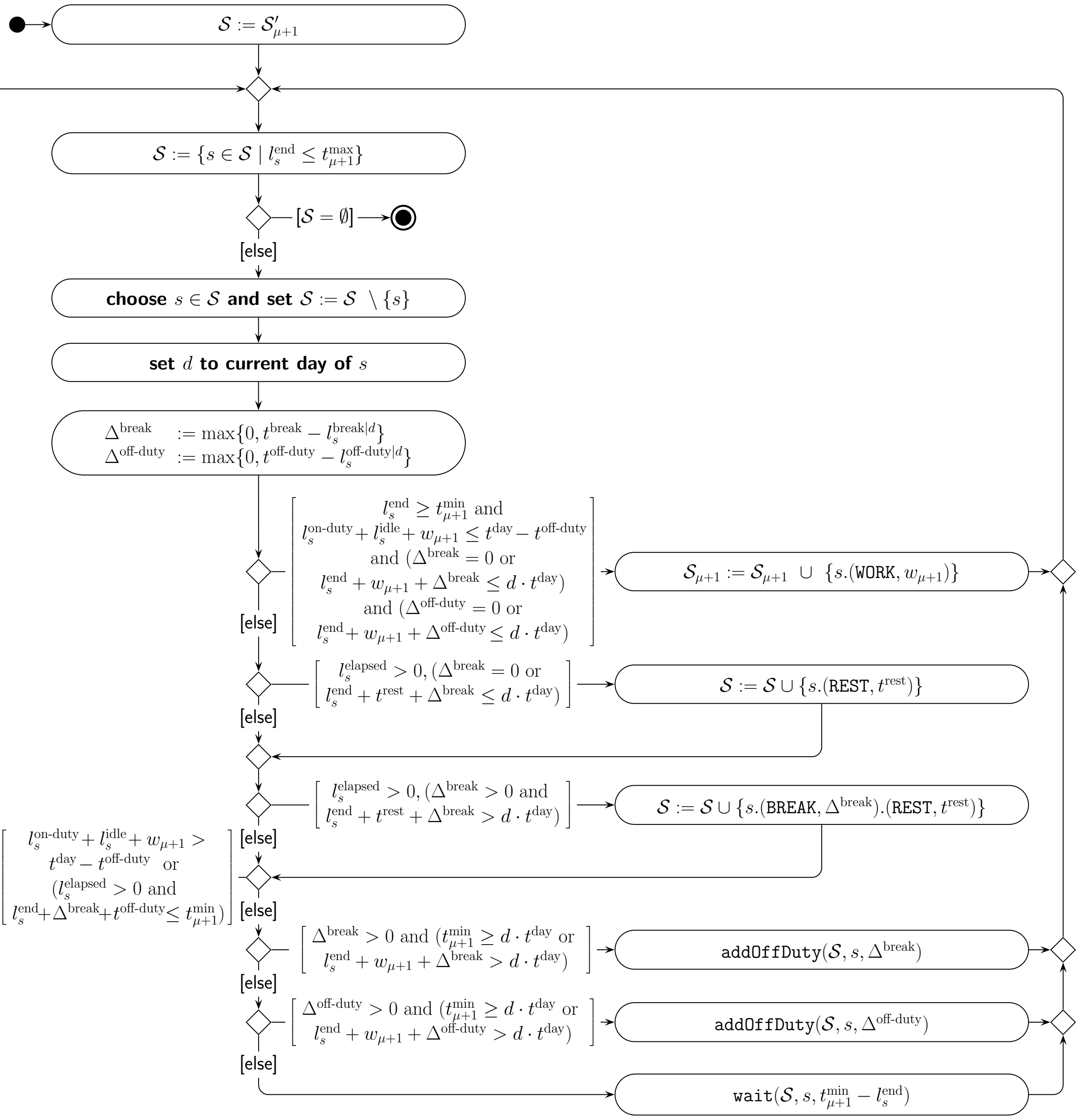

Figure 3: Method for scheduling stationary activities after arrival at node $n_{\mu+1}$

the schedule before the work activity can be scheduled. In either case the methods generates new schedules by increasing the amount of off-duty time using the method $\texttt{addOffDuty}(\cdot,\cdot,\cdot)$. After increasing the accumulated amount of off-duty time the method continues with the next loop.

If neither of these cases hold, the methods generates new schedules by increasing the completion time of the schedule to $t^{\text{min}}_{\mu+1}$ using the method $\texttt{wait}(\cdot,\cdot,\cdot)$. After that the method continues with the next loop.

In each iteration we remove dominated schedules from $\mathcal{S}_\mu$ in order to reduce the computational effort required by the scheduling method.

7 Solution Approaches

In this section we present solution approaches using the framework presented in the previous section. The first solution approach is a heuristic in which the methods $\texttt{addOffDuty}(\mathcal{S}, s, \Delta)$ and $\texttt{wait}(\mathcal{S}, s, \Delta)$ simply add a schedule $s.(\texttt{BREAK}, \max\{\Delta, l^{\text{length}}_s\})$ to the set $\mathcal{S}$. If $l^{\text{elapsed}}_s = 0$, every other schedule that could be generated would be dominated by $s.(\texttt{BREAK}, \Delta)$. However, if $l^{\text{elapsed}}_s > 0$ the time elapsed since the end of the off-duty period commencing with the last rest period increases when appending a break period. Thus, it may be beneficial to insert a break after the last rest period, instead of appending a break period. Figure 4 illustrates another heuristic method for $\texttt{addOffDuty}(\mathcal{S}, s, \Delta)$ which can generate two alternative schedules: the first schedule is the schedule which is obtained by appending a break period; the second schedule is a schedule obtained by inserting off-duty time after the last rest period. The method first determines the current day d of schedule s. If $l^{\text{end}}_s + \Delta > d \cdot t^{\text{day}}$ then it is impossible to increase the amount of off-duty time on day d by the required amount and the method terminates. Otherwise, the method generates a new schedule by appending a break activity of duration $\max\{\Delta, l^{\text{length}}_s\}$ to schedule s and includes the new schedule in the set $\mathcal{S}$. If $l^{\text{elapsed}}_s = 0$, there is no better way to increase the amount of off-duty time and the method terminates.

Otherwise, it sets $s' := s$ and $\Delta' := \Delta$ and tries to insert off-duty time after the last rest. If $l^{\text{push}|\texttt{R}}_{s'} = 0$ or $l^{\text{end}}_s + \Delta > d \cdot t^{\text{day}}$ the method terminates because either it cannot insert off-duty time after the last rest or there is not enough time to add enough off-duty time on the current day.

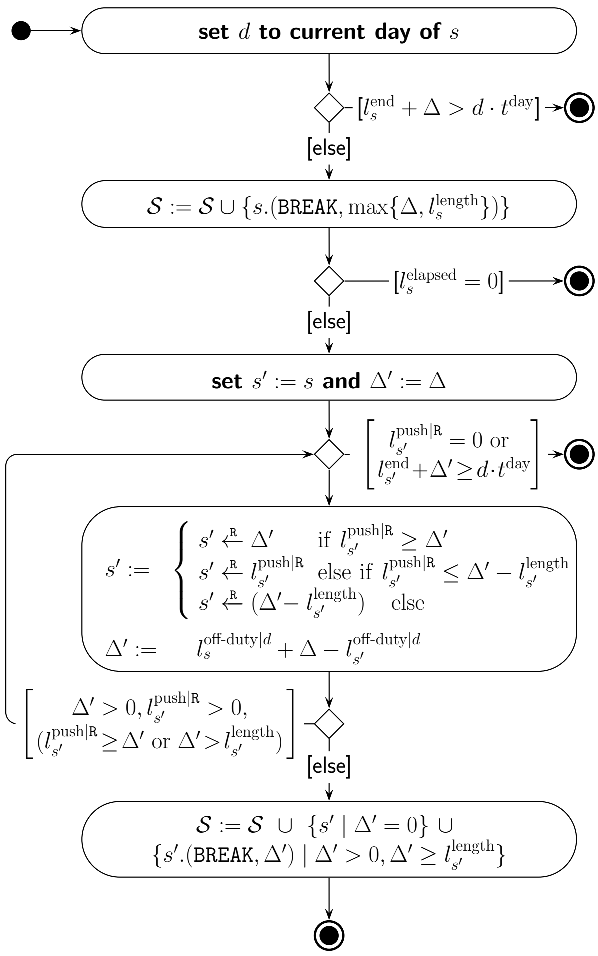

Figure 4: Heuristic method $\texttt{addOffDuty}(\mathcal{S}, s, \Delta)$

Otherwise, it inserts a break period of duration Δ' after the last rest if $l_{s'}^{\text{push}|\text{R}} \geq \Delta'$. If $l_{s'}^{\text{push}|\text{R}} < \Delta'$ it is not possible to insert the required amount of off-duty time. The method inserts a break period of duration $l_{s'}^{\text{push}|\text{R}}$ after the last rest if $l_{s'}^{\text{push}|\text{R}} \leq \Delta' - l_{s'}^{\text{length}}$. Otherwise, it inserts a break period of duration $\Delta' - l_{s'}^{\text{length}}$

The method updates Δ' to the remaining amount of off-duty time that needs to be added. Note that inserting off-duty time after the last rest may not always increase the amount of off-duty time on the current day, e.g. if the last rest ends on the previous day and the last activity on the previous day is a driving period, then inserting a break after the last rest might simply push the driving period into the current day. If $\Delta' > 0$ further off-duty time must be added on the current day. If $l_{s'}^{\text{push}|\mathtt{R}} > \Delta'$ or $\Delta' > l_{s'}^{\text{length}}$ and $l_{s'}^{\text{push}|\mathtt{R}} > 0$, the method repeats the previous steps in order to increase the amount of off-duty time on the current day. Otherwise, the method inserts s' to the set $\mathcal{S}$ if no further off-duty time is required, or it inserts $s'.(\mathtt{BREAK}, \Delta')$ to the set $\mathcal{S}$ if a positive amount of off-duty time must still be added and $\Delta' \geq l_{s'}^{\text{length}}$.

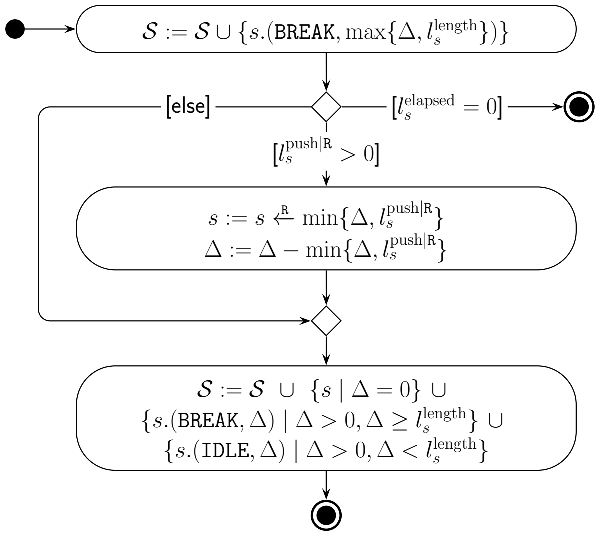

Figure 5: Heuristic method $\mathtt{wait}(\mathcal{S}, s, \Delta)$

Figure 5 illustrates a similar heuristic method for $\mathtt{wait}(\mathcal{S}, s, \Delta)$. The method first generates a new schedule by appending a break activity of duration $\max\{\Delta, l_s^{\text{length}}\}$ to schedule s and includes the new schedule in the set $\mathcal{S}$. If $l_s^{\text{elapsed}} = 0$, there is no better way to increase the completion time of the schedule and the method terminates. Otherwise, it inserts a break period of duration $\min\{\Delta, l_s^{\text{push}|\mathtt{R}}\}$ after the last rest in the schedule s and updates the remaining amount by which the completion time must be increased. The method inserts s to the set $\mathcal{S}$ if the completion time is sufficiently increased, inserts $s.(\mathtt{BREAK}, \Delta)$ to the set $\mathcal{S}$ if the completion time must still be increased by a positive amount of $\Delta \geq l_s^{\text{length}}$, or inserts $s.(\mathtt{IDLE}, \Delta)$ to the set $\mathcal{S}$ if the completion time must still be increased by a positive amount of $\Delta < l_s^{\text{length}}$. Thereafter, the method terminates.

The heuristic approaches presented above only generate schedules by appending off-duty time to a schedule or by trying to insert as much off-duty time after the last rest as possible. In order to solve the CAN-TDSP, however, every alternative of modifying a schedule which does not result in a schedule dominated by others needs to be generated.

Figure 6 illustrates an enumerative method for $\texttt{addOffDuty}(\mathcal{S}, s, \Delta)$ which generates all alternative non-dominated schedules in which the amount of off-duty time on the current day is Δ higher than in schedule s. The enumerative method starts with the same steps as the heuristics method illustrated in Figure 4.

After setting $s' := s$ and $\Delta' := \Delta$ it iteratively inserts break periods of $\frac{1}{4}$ hour after the last rest or break in order to generate all alternative non-dominated schedules. If $\Delta' < l_{s'}^{\text{length}}$ and $\Delta' \leq l_{s'}^{\text{push}|\texttt{B}}$, the method generates a new schedule $s'' := s' \overset{\texttt{B}}{\leftarrow} \Delta'$ and adds this schedule to $\mathcal{S}$ if $l_{s''}^{\text{off-duty}|d} = l_{s}^{\text{off-duty}|d} + \Delta$. If $l_{s''}^{\text{off-duty}|d} < l_{s}^{\text{off-duty}|d} + \Delta$ then this new schedule is discarded because appending a break period of duration $l_{s'}^{\text{length}}$ would be the better alternative.

If $l_{s'}^{\text{push}|\texttt{R}} < \frac{1}{4}$ the method terminates because all non-dominated alternatives have been generated. Otherwise, the method sets $s' := s' \overset{\texttt{R}}{\leftarrow} \frac{1}{4}$ and updates Δ' to the remaining off-duty time that must be added to s'. If $l_{s'}^{\text{off-duty}|d} = l_{s}^{\text{off-duty}|d} + \Delta$ then s' is added to $\mathcal{S}$. If $\Delta' = 0$ or $l_{s'}^{\text{end}} + \Delta' > d \cdot t^{\text{day}}$ the method terminates, because no other non-dominated schedules can be generated. Otherwise, the method generates a new schedule by appending a break activity of duration Δ' to schedule s' and includes the new schedule in the set $\mathcal{S}$ if $\Delta' \geq l_{s}^{\text{length}}$. Then, the method continues with the next iteration.

Figure 7 illustrates an enumerative method for $\texttt{wait}(\mathcal{S}, s, \Delta)$ which generates all non-dominated alternative schedules which can be generated in order to increase the completion time of schedule s by a positive value Δ. The method first generates a new schedule by appending a break activity of duration $\max\{\Delta, l_{s}^{\text{length}}\}$ to schedule s and includes the new schedule in the set $\mathcal{S}$. If $l_{s}^{\text{elapsed}} = 0$, there is no better way to increase the completion time of the schedule and the method terminates.

Otherwise, it sets $s' := s$ and $\Delta' := \Delta$ and iteratively inserts break periods of $\frac{1}{4}$ hour until all non-dominated alternative schedules are determined. If $0 < \Delta' < l_{s'}^{\text{length}}$ then a $\frac{1}{4}$ hour must still be added to the completion time. As the minimum duration of a break period to be appended to the schedule is $\frac{1}{2}$ hour, it could be beneficial to insert some break time after the last break in the schedule. If furthermore $\Delta' \leq l_{s'}^{\text{push}|\texttt{B}}$ the method generates a new schedule by inserting a break period of $\frac{1}{4}$ hour

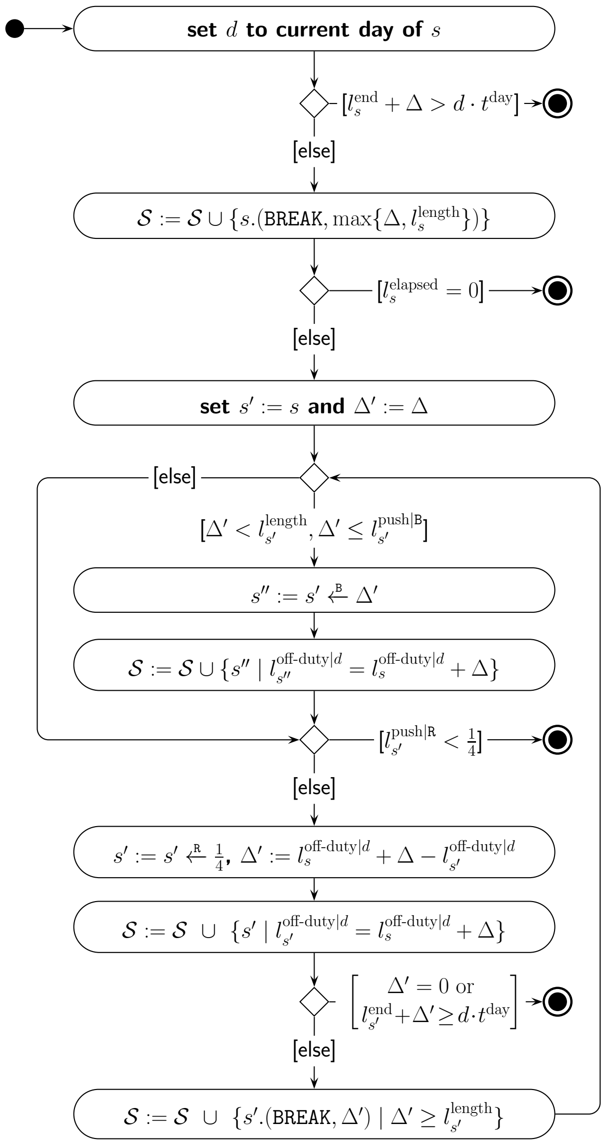

Figure 6: Enumerative method $\texttt{addOffDuty}(\mathcal{S}, s, \Delta)$

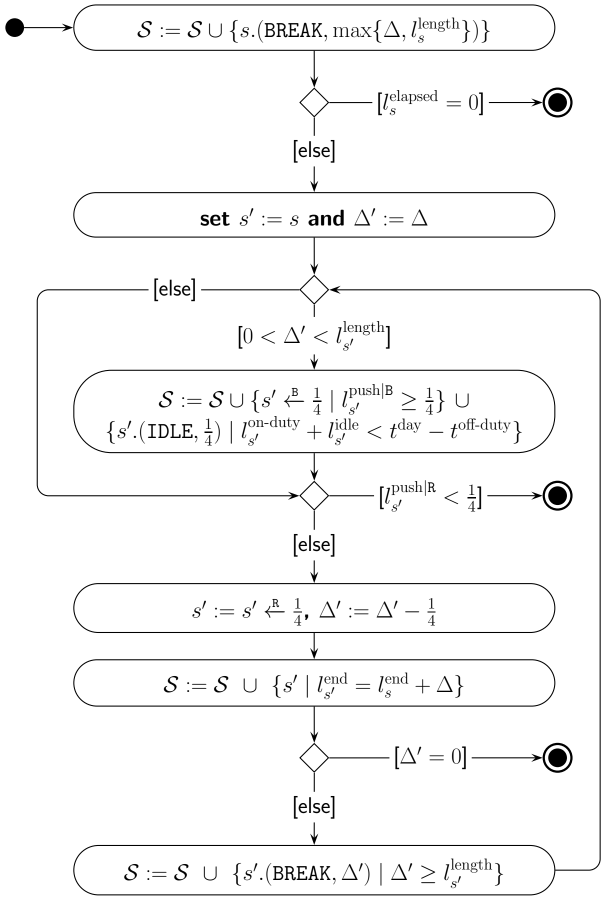

Figure 7: Enumerative method $\mathtt{wait}(\mathcal{S}, s, \Delta)$

after the last break in s' and adds the new schedule to $\mathcal{S}$. If $l_{s'}^{\text{on-duty}} + l_{s'}^{\text{idle}} < t^{\text{day}} - t^{\text{off-duty}}$ the method generates a new schedule by appending an idle period of $\frac{1}{4}$ hour to s' and adds the new schedule to $\mathcal{S}$.

If $l_{s'}^{\text{push}|\mathtt{R}} < \frac{1}{4}$ the method terminates because all non-dominated alternatives have been generated. Otherwise, the method sets $s' := s' \overset{\mathtt{R}}{\leftarrow} \frac{1}{4}$ and decreases Δ' by $\frac{1}{4}$. If $l_{s'}^{\text{end}} = l_s^{\text{end}} + \Delta$ then the updated schedule is added to $\mathcal{S}$. If $\Delta' = 0$ then the completion time has been sufficiently increased and the method terminates. If $\Delta' \geq l_s^{\text{length}}$, the method generates a new schedule by appending a break activity of duration Δ' to schedule s' and includes the new schedule in the set $\mathcal{S}$. Then, the method continues with the next iteration.

8 Computational Experiments

In order to evaluate the scheduling methods presented in this paper we generated three sets of benchmark instances for a planning horizon starting on Monday 0.00 AM and ending on Friday 11.59 PM. In all benchmark sets one hour of work time shall be conducted at each location in the tour. The driving time between two subsequent locations is randomly set to a value between 2 and 10 hours for the first set, between 10 and 20 hours for the second set, and between 2 and 20 hours for the third set. Assuming an average speed of 75 km/h, this implies that the distance between two subsequent locations ranges from 150 km to 1500 km. The duration of the time windows of the locations are randomly set to a value between 1 and 12 hours. The start times of the time windows are randomly set to a time between 15 minutes and 6 hours after the earliest departure at the previous location increased by the pure driving time multiplied by 1.5. All instances generated have an accumulated on-duty time of 70 hours or less.

Table 2 shows the results of our computational experiments. In the table CAN* refers to the exact approach using the enumerative methods illustrated in Figures 6 and 7. CAN1 refers to the approach using the heuristic which only appends break periods, and CAN2 refers to the approach using the heuristics illustrated in Figures 4 and 5. In order to compare the results of these methods with other existing approaches we adapted the exact approach for the U.S. Truck Driver Scheduling Problem (US-TDSP) presented by Goel and Kok (2010) to the Canadian case. For this, however, we had to relax conditions (9) to (13) of the CAN-TDSP. In order to tackle the remaining conditions only small changes to the approach had to be made. CAN-R8 refers to the approach for the relaxed problem using the relevant parameters of Table 1, and CAN-R10 refers to the approach for the relaxed problem using the same parameters, except for t^{rest} which is set to 10 hours. As CAN-R8 and CAN-R10 do not consider all constraints, some of the truck driver schedules found by these methods may not be feasible for the CAN-TDSP. In the table we present the number of feasible schedules for the relaxed problem. For further comparison, we also ran experiments with the method presented by Goel and Kok (2010) in order to determine for how many instances a feasible schedule for the US-TDSP exists.

We can see in Table 2 that the heuristic approaches CAN1 and CAN2 are very fast and have similar running times as the CAN-R8 and CAN-R10 approaches which are known to have a worst

Algorithm	**Driving Time**	**Instances**	**Feasible**	**Total CPU Time**
CAN*	2-10	991	**878**	179.34s
CAN1	2-10	991	805	0.68s
CAN2	2-10	991	858	1.04s
CAN-R8	2-10	991	*934*	0.55s
CAN-R10	2-10	991	*609*	0.42s
US	2-10	991	*434*	0.36s
CAN*	10-20	837	**744**	526.08s
CAN1	10-20	837	695	0.28s
CAN2	10-20	837	**744**	0.39s
CAN-R8	10-20	837	*830*	0.26s
CAN-R10	10-20	837	*697*	0.22s
US	10-20	837	*221*	0.18s
CAN*	2-20	872	**734**	907.95s
CAN1	2-20	872	688	0.37s
CAN2	2-20	872	729	0.58s
CAN-R8	2-20	872	*847*	0.33s
CAN-R10	2-20	872	*616*	0.27s
US	2-20	872	*297*	0.23s

Table 2: Results

case complexity of $O(\lambda^2)$. The CAN2 heuristic finds a feasible schedule, if one exists, for almost 99 percent of the instances. For the set of benchmark instances with driving times between 10 and 20 hours, the CAN2 heuristic even succeeds in finding a feasible schedule for all instances for which a feasible schedule exists. The exact approach CAN* requires much more time, which is not surprising as the number of alternatives explored by the method is significantly higher. It must be noted that the CAN* approach makes use of some additional dominance criteria which are not reported in this paper. These criteria can be easily derived from the dominance criteria presented in this paper, but have been omitted for reasons of conciseness. Without an extensive usage of dominance criteria the exact approach would not be able to terminate in a comparable amount of time.

As the CAN-R8 method solves the relaxed problem with the same parameters, we know that if no truck driver schedule is found by the CAN-R8 method, then no feasible schedule exists for the CAN-TDSP. The number of instances for which the CAN-R8 method finds a feasible schedule is much higher compared to the other methods. However, as conditions (9) to (13) are relaxed many of the schedules found by the CAN-R8 method are infeasible for the original problem. The CAN-R10 method represents an alternative approach in which each rest period must have a duration of at least 10 hours. By this, the amount of 10 hours of off-duty time a day which is required in the original problem is implicitly considered. The CAN-R10 method finds a feasible schedule for significantly less instances than the other approaches. Furthermore, despite increasing the duration of rest periods, the schedules which are feasible for the relaxed problem may not necessarily be feasible for the original problem. These results indicate, that determining truck driver schedules complying with Canadian regulations is not simply a matter of adapting approaches designed for U.S. hours of service regulations. Instead, the additional constraints imposed by Canadian regulations must be explicitly considered.

When comparing the number of instances for which a feasible schedule exists complying with Canadian regulations compared to U.S. regulations, we can see that Canadian regulations are much more permissive. Even though, the maximum amount of driving on a day is higher according to U.S. hours of service regulations, the higher amount of continuous off-duty time that needs to be taken appears to reduce the degree of freedom when generating truck driver schedules. As a result of the lower degree of freedom, a feasible schedule complying with U.S. regulations can be found for significantly fewer instances.

9 Conclusions

In this paper we study the Canadian Truck Driver Scheduling Problem (CAN-TDSP). We present a solution framework that can be used to solve the CAN-TDSP using heuristic or exact approaches. The heuristics presented in this paper can find feasible truck driver schedules for up to 99 percent of the instances for which the exact approach showed that a feasible schedule exists. Our heuristic approaches are very fast and only require little more time than an adaptation of the scheduling approach presented by Goel and Kok (2010) which is known to have a worst case complexity of $O(\lambda^2)$. As the adaptation of the scheduling approach presented by Goel and Kok (2010) requires to relax some of the constraints of the CAN-TDSP, the solution quality is much lower than the solution quality obtained by the heuristics specifically developed for the CAN-TDSP. The exact method presented requires much more time to solve and is not competitive in terms of running time. However, it must be noted that by using effective dominance criteria we were able to reduce the amount of time required to solve the CAN-TDSP to a small fraction of the time that would be required without usage of dominance criteria.

Our results indicate, that from a scheduling perspective, Canadian regulations are much more permissive than U.S. hours of service regulations which are currently being revised due to concerns about their impact on road safety. It must be clearly stated that this paper does not analyse the impact of Canadian regulations on road safety. Although this would be an interesting question it is out of scope of this paper. Our computational experiments give an indication of how likely it is, that truck drivers can comply with the regulation if they are given a sequence of tasks that has been generated without considering the regulation. Furthermore, we provide methods that can be used by motor carriers to ensure that all tasks assigned to their truck drivers can be executed without violating the regulation. The approaches can contribute to increased road safety if they are used to verify that drivers can comply with the regulations when generating delivery schedules and routes.

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.

Canadian Council of Motor Transport Administrators. Commercial Vehicle Drivers Hours of Service Regulations - Application Guide. ISBN 0-921795-82-3, 2007.

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

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

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

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

A. Goel and L. Kok. Truck Driver Scheduling in the United States. Working Paper, MIT-Zaragoza International Logistics Program, 2010.

A. L. Kok, C. M. Meyer, H. Kopfer, and J. M. J. Schutten. A Dynamic Programming Heuristic for the Vehicle Routing Problem with Time Windows and European Community Social Legislation . *Transportation Science*, 44(4):442–454, 2010.

A. T. McCartt, L. A. Hellinga, and M. G. Solomon. Work schedules of long-distance truck drivers before and after 2004 hours-of-service rule change. *Traffic Injury Prevention*, 9(3):201–210, 2008. doi: 10.1080/15389580802040287.

M. M. Mitler, J. C. Miller, J. J. Lipsitz, J. K. Walsh, and C. D. Wylie. The sleep of long-haul truck drivers. *New England Journal of Medicine*, 337(11):755–762, 1997. doi: 10.1056/NEJM199709113371106.

E. Prescott-Gagnon, G. Desaulniers, M. Drexl, and L.-M. Rousseau. European Driver Rules in Vehicle Routing with Time Windows. *Transportation Science*, 44(4):455–473, 2010. doi: 10.1287/trsc.1100.0328.

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

Transport Canada. Commercial Vehicle Drivers Hours of Service Regulations. SOR/2005-313, 2005. URL `http://laws.justice.gc.ca/en/SOR-2005-313/`.

A. Williamson, S. Sadural, A. Feyer, and R. Friswell. Driver Fatigue: A Survey of Long Distance Heavy Vehicle Drivers in Australia. Information Paper CR 198, National Road Transport Commission, 2001.

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

Appendix

Proof of Lemma 1. Let $\bar{s}''$ denote the partial schedule obtained by applying the first operator move to s''. If $\bar{s}'' = s''.a$ for some activity a let us set $\bar{s}' := s'.a$. If $\bar{s}''$ complies with time window constraints and the regulation, the same applies for $\bar{s}'$. Thus, all of the conditions of the lemma maintain valid for $\bar{s}'$ and $\bar{s}''$. If $\bar{s}'' = s'' \overset{\mathtt{R}}{\leftarrow} \Delta$ for some $\Delta > 0$ let us set $\bar{s}' := s' \overset{\mathtt{R}}{\leftarrow} \Delta$. Otherwise, we have $\bar{s}'' = s'' \overset{\mathtt{B}}{\leftarrow} \Delta$ for some $\Delta > 0$ and we set $\bar{s}' := s' \overset{\mathtt{B}}{\leftarrow} \Delta$. We know that $\bar{s}'$ complies with time window constraints and the regulation, because $l_{s'}^{\text{push}|\mathtt{R}} \geq l_{s''}^{\text{push}|\mathtt{R}} \geq \Delta$ or $l_{s'}^{\text{push}|\mathtt{B}} \geq l_{s''}^{\text{push}|\mathtt{B}} \geq \Delta$ and all of the conditions of the lemma maintain valid for $\bar{s}'$ and $\bar{s}''$. Either $\bar{s}'$ is a feasible schedule or we can iteratively replace s' by $\bar{s}'$ and s'' by $\bar{s}''$ and do the same for the next operator move until a feasible schedule is found. □

Proof of Lemma 2. If $t^{\text{break}} > l_{s'}^{\text{break}|d'}$ we can set $\bar{s}' := s'.(\mathtt{BREAK}, t^{\text{break}} - l_{s'}^{\text{break}|d}).(\mathtt{REST}, t^{\text{rest}})$ $.(\mathtt{BREAK}, l_{s''}^{\text{end}} - l_{s'}^{\text{end}} - (t^{\text{break}} - l_{s'}^{\text{break}|d}) - t^{\text{rest}})$. Otherwise we can set $\bar{s}' := s'.(\mathtt{REST}, t^{\text{rest}})\ .(\mathtt{BREAK}, l_{s''}^{\text{end}} - l_{s'}^{\text{end}} - t^{\text{rest}})$. Then, we have $l_{\bar{s}'}^{\text{end}} = l_{s''}^{\text{end}}$, $l_{\bar{s}'}^{\text{elapsed}} = 0$, $l_{\bar{s}'}^{\text{push}|\mathtt{R}} \geq l_{s''}^{\text{push}|\mathtt{R}}$, and $l_{\bar{s}'}^{\text{push}|\mathtt{B}} \geq l_{s''}^{\text{push}|\mathtt{B}}$. Let us denote with $\bar{d}$ the current day of $\bar{s}'$. If $\bar{d} = d$ we have $l_{\bar{s}'}^{\text{drive}|\bar{d}} \leq l_{s''}^{\text{drive}|\bar{d}}$. Otherwise, we have $l_{\bar{s}'}^{\text{drive}|\bar{d}} = 0$. Furthermore, we have $l_{\bar{s}'}^{\text{break}|\bar{d}} \geq l_{s''}^{\text{break}|\bar{d}}$ or $l_{\bar{s}'}^{\text{break}|\bar{d}} \geq t^{\text{break}}$, and $l_{\bar{s}'}^{\text{off-duty}|\bar{d}} \geq l_{s''}^{\text{off-duty}|\bar{d}}$ or $l_{\bar{s}'}^{\text{off-duty}|\bar{d}} \geq t^{\text{off-duty}}$. Thus, for all operator moves needed to transform s'' into a feasible schedule, the partial schedules obtained by applying the corresponding moves to $\bar{s}'$ are feasible. Dominance of $\bar{s}'$ over s'' can, therefore, be shown analogously to Lemma 1. □