CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Long-Haul Vehicle Routing and Scheduling with Idling Options

Çağri Koç
Ola Jabali
Gilbert Laporte

September 2016

CIRRELT-2016-47

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Long-Haul Vehicle Routing and Scheduling with Idling Options

Çağri Koç[1,2,*], Ola Jabali[1,3], Gilbert Laporte[1,2]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Decision Sciences, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[3] Department of Logistics and Operations Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Abstract. This paper introduces the Vehicle Routing and Truck Driver Scheduling Problem with Idling Options (VRTDSP-IO), an extension of the long-haul vehicle routing and truck driver scheduling problem with a more comprehensive objective function that accounts for routing cost, driver cost and idling cost, i.e., the cost associated with energy supply used to maintain drivers' comfort when the vehicle is not moving. For the idling cost, we consider Electrified Parking Space (EPS) and Auxiliary Power Unit (APU) usage costs. The use of EPSs or APUs avoids keeping the vehicle engine running while the vehicle is not moving. We develop a multi-start matheuristic algorithm that combines adaptive large neighborhood search and mixed integer linear programming. We present extensive computational results on instances derived from the Solomon test-bed.

Keywords: Long-haul vehicle routing, truck driver scheduling, engine idling, hours of service regulations, matheuristic, multi-start.

Acknowledgements. The authors gratefully acknowledge funding provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants 2015-06189 and 436014-2013.

---

* Corresponding author: Cagri.Koc@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2016

1 Introduction

In long-haul freight transportation, a fleet of vehicles must visit a set of customers over several days, quite often for an entire week. Because customer locations normally close at night, these will typically have multiple time windows over the planning horizon. In addition, the drivers are subject to Hour of Service (HOS) regulations which specify minimum break or rest durations in relation to the driving or to the on-duty times. There exists a rich literature on the combined routing and scheduling problem encountered in long-haul freight transportation. An important yet less studied issue arising in this context is related to vehicle idling. When drivers take breaks or rests or when they service customers, they often leave their engine running in order to keep their vehicle at a comfortable temperature or to benefit from amenities such as television. According to Rahman et al. (2013), long-haul trucks idle between six and 16 hours per day. This practice is not only expensive, but it damages the vehicles and is highly harmful to the environment. In a recent study, Koç et al. (2016b) have shown that Electric Parking Spaces (EPSs) and Auxiliary Power Units (APUs) constitute interesting idling options that can be used instead of engine idling. EPSs allow vehicles to be plugged into an electrical power source, whereas APUs are fuel-based pieces of equipment added to the vehicle, that generate sufficient power to maintain an adequate comfort level in the vehicle, but consumes far less fuel than engine idling. Given a vehicle route and considering the fixed and variable costs of EPSs and APUs, Koç et al. (2016b) have investigated the optimal use of idling options. The aim of this paper is to develop a multi-start matheuristic to solve the combined routing, scheduling and idling option selection problem. Before we proceed with our study, we briefly review the literature on each of these three components.

1.1 Literature review

The long-haul Truck Driver Scheduling Problem (TDSP) under the HOS regulations, which does not consider routing decisions, has been studied by several researchers in recent years. Archetti and Savelsbergh (2009) developed an algorithm to solve the problem of sequencing full truckload requests that have a dispatch window at the origin. The algorithm generates a feasible schedule in polynomial time if one exists. Goel (2012a) later studied a variant of the TDSP with the aim of minimizing total duration under the United States (US) and European Union (EU) regulations. The author proposed a mixed integer linear programming formulation and a dynamic programming algorithm. Goel and Kok (2012) developed an algorithm to solve the similar problem of Goel (2012a) in which each customer location must be visited within one of several time windows. The complexity of the algorithm is similar to that of the single window case. Goel and Rousseau (2012) and Goel (2012b) studied the TDSP under the Canadian HOS regulations. Goel and Rousseau (2012) proposed an exact algorithm and two heuristics, and Goel (2012b) developed an iterative dynamic programming algorithm. Goel (2012c) and Goel et al. (2012) considered the TDSP under the Australian HOS regulations. Goel (2012c) presented a mixed integer programming formulation and valid inequalities, while Goel et al. (2012) developed four heuristics and an exact method.

The joint routing and scheduling problem under the HOS regulations, called the Vehicle Routing and Truck Driver Scheduling Problem (VRTDSP), has been considered by several authors. Xu et al. (2003) were the first to study this combined problem. Goel (2009) considered the EU regulations. For the same regulations, Kok et al. (2010) developed a hybrid method that integrates a basic break scheduling procedure within a dynamic programming framework. Prescott-Gagnon et al. (2010) proposed a large neighborhood search algorithm based on a column generation heuristic. Rancourt et al. (2013) put forward a hybrid tabu search algorithm which includes several scheduling procedures to solve the VRTDSP under the US regulations with a heterogeneous fleet of vehicles. Goel and Vidal (2014) developed a unified genetic algorithm for the EU, US, Canadian and Australian HOS regulations. Goel and Irnich (2016) later developed a branch-and-price algorithm which is the first exact method for the VRTDSP.

More recently, Koç et al. (2016b) introduced, modeled and solved exactly the Truck Driver Scheduling Problem with Idling Options (TDSP-IO) within a long-haul transportation context under the US HOS regulations. The problem has a more comprehensive objective function than that of the TDSP since it accounts for the cost of fuel consumption along with the costs of drivers and idling options. In addition, unlike the papers just reviewed, it is no longer assumed that the trucks can stop anywhere along their route. Instead, the rests and breaks imposed by the HOS regulations must be taken at locations belonging to a prespecified set. By performing extensive sensitivity analyses, the authors quantified the advantages of using EPSs and APUs relative to engine idling, both in terms of economic and environmental benefits. One of their main conclusions is that it is worth investing in both EPS and APU equipment.

1.2 Scientific contributions and structure of the paper

This paper makes two main scientific contributions. First, it introduces the Vehicle Routing and Truck Driver Scheduling Problem with Idling Options (VRTDSP-IO) as an extension of the VRTDSP and of the TDSP-IO, using a comprehensive objective function that minimizes routing costs, driver costs, as well as EPS and APU usage costs. The second contribution is the development of a multi-start matheuristic capable of efficiently solving the problem. The algorithm successfully combines adaptive large neighborhood search (ALNS) with an exact scheduling and idling optimization component.

The remainder of this paper is structured as follows. Section 2 provides some technical background for the problem, which is then formally defined and modeled in Section 3. Section 4 contains a description of the matheuristic. Computational results are presented in Section 5, followed by conclusions in Section 6.

2 Technical background

We first briefly describe the two idling alternatives (EPS and APU) that have been shown to be preferable to engine idling (Koç et al., 2016b), followed by the US HOS regulations and their pa-

rameters.

2.1 Idling options

Following the conclusions of Koç et al. (2016b), we assume in this study that each vehicle is equipped with EPS and APU apparatus, so that these two options are effectively available and their fixed costs can be disregarded.

2.1.1 Electrified Parking Space (EPS) idling

EPSs provide power for heating, ventilation, air conditioning and other amenities without idling the engine and allow truck drivers to switch off their engines. The US National Renewable Energy Laboratory (NREL) (NREL, 2016) publicly shares EPS data of current EPS owner companies (EnviroDock, 2016; Shorepower, 2016), in order to inform heavy-duty truck companies and truck drivers about the EPS locations (DOE, 2016a). Trucks making use of an EPS require an on-board equipment which includes an inverter to convert 120-volt power, electrical equipment, and hardware, so that they can plug into an off-board outlet. The trucking company owns and maintains the on-board equipment. The EPS equipment fixed cost is around \$2500 and its variable cost is around \$1.00/h (Argonne National Laboratory, 2015).

2.1.2 Auxiliary Power Unit (APU) idling

Drivers can use an APU, which is powered by diesel and provides on-board power for climate control and electrical devices. Similar to engine idling an APU can be used at Interstate Rest Areas (IRAs) and at customer locations (Carrier, 2016). Many IRAs are available on US highways (US Rest Areas, 2016). APUs have an acquisition cost of around \$10,000 and a variable cost of around \$0.63/h (Argonne National Laboratory, 2015).

2.2 The United States Hours of Service regulations

In the US, the HOS regulations (FMCSA, 2014) distinguish between *on-duty time* and *off-duty time*. *On-duty time* means that a truck driver is working. It includes driving time, which is the time spent at the driving controls of a vehicle in operation, as well as the time needed for other activities such as waiting for service, supervising, loading and unloading, and handling paperwork. *Off-duty time* consists of break periods and rest periods during which truck drivers have no obligation to perform any work. Table 1 summarizes the parameter values of the regulations.

3 Formal problem definition and mathematical formulations

Given a homogeneous fleet of vehicles, the VRTDSP-IO aims to determine a set of routes, and their corresponding schedules, which include the choice of idling options. The solution to the problem ensures that each customer demand is satisfied within one of its time windows, all vehicles start and end their routes at the depot, the total load assigned to a vehicle does not exceed its capacity, and the HOS regulations are satisfied. For given routes, the problem then consists

Table 1: Parameters imposed by the US HOS regulations.

Notation	Value (h)	Description
t^{duty}	60	The maximal cumulative on-duty hours during seven consecutive days.
$t^{driving}$	11	The maximal cumulative driving hours between two rest periods.
t^{rest}	14	The maximal cumulative on-duty hours since the end of the last rest period.
t^{break}	8	The maximal cumulative on-duty hours since the end of the last rest or break period.
$t^{minrest}$	10	The minimal duration of a rest period to regain driving time.
$t^{minbreak}$	0.5	The minimal duration of a break period to regain driving time.

of deciding when the truck driver will drive, serve a customer, rest or break at an IRA or at an EPS. The main objective of the VRTDSP-IO is to minimize the *total cost* which is made up of three main components: *routing cost*, *driver cost* and *idling cost*. The routing cost is proportional to the traveled distance and corresponds to the cost of fuel while the vehicle is moving. The driver cost corresponds to the driver's wages while on-duty; since the driving and service times on a given route are constant, the only part of the driver cost that can be optimized is related to the waiting time before service at customer locations. The idling cost is associated with EPS and APU usage while the driver is off-duty (and not paid) and the vehicle is not moving.

The VRTDSP-IO is defined on a complete directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A} = \{(i,j) : i,j \in \mathcal{N}, i \neq j\}$ is the set of arcs. The set of nodes is partitioned into $\mathcal{N} = \{\mathcal{N}_0, \mathcal{N}_c\}$, where $\mathcal{N}_0 = \{0, n+1\}$ contains two copies of the depot at which each route starts and ends, and $\mathcal{N}_c = \{1, \ldots, n\}$ is the set of customers. In addition, we define two other set of nodes: the set $\mathcal{N}_e$ of EPSs and the set $\mathcal{N}_a$ of IRAs appearing on the arcs of $\mathcal{A}$. The nodes of $\mathcal{N}_e$ and $\mathcal{N}_a$ appearing on different arcs are not directly connected to each other. Each arc $(i,j) \in \mathcal{A}$ has a nonnegative distance c_{ij}. The driving time in hours from node i to node j is denoted by d_{ij}. Let Q denote the capacity of the vehicle. Each customer $i \in \mathcal{N}_c$ has a positive demand q_i, a service time s_i, and each node of $\mathcal{N}$ has an ordered set T_i of time windows, all expressed in hours. The service at a customer must begin within one of its time windows, and must be completed without interruption. The τ^{th} time window ($\tau \in T_i$) at node $i \in \mathcal{N}$ is denoted by the interval $[t_{i\tau}^{min}, t_{i\tau}^{max}]$. If a vehicle arrives before the opening of the first available time window, it has to wait. It must also arrive before the closing of the selected time window because otherwise it will have to wait until the opening of the next available time window. The time horizon is denoted by $t^{horizon}$, typically 168 h. We assume that the two copies of depot have very wide time windows $[t_{i1}^{min} = 0, t_{i1}^{max} = t^{horizon}]$.

3.1 Set partitioning model

We formulate the VRTDSP-IO as a Set Partitioning Problem (SPP) on $\mathcal{G}$. Let $\mathcal{R}$ be the set of all feasible routes. Let θ_r be a binary variable equal to 1 if route r is selected, and equal to 0 otherwise. Let C_r be the total cost of route r. Let a_{ir} be a binary parameter equal to 1 if a customer i is on

route $r \in \mathcal{R}$, and equal to 0 otherwise. The SPP is then:

$$\text{Minimize} \quad \sum_{r \in \mathcal{R}} C_r \theta_r \tag{1}$$

subject to

$$\sum_{r \in \mathcal{R}} a_{ir} \theta_r = 1 \qquad i \in \mathcal{N}_c \tag{2}$$

$$\theta_r \in \{0, 1\} \qquad r \in \mathcal{R}. \tag{3}$$

The objective function (1) computes the total cost of the solution. Constraints (2) guarantee that each customer is assigned to exactly one route. Constraints (3) define the domain of the decision variables. Next, we detail the computation of the optimal driver and idling costs on a given route r.

3.2 Computation of the optimal driver and idling costs of a given route

The optimal driver and idling cost on a given route r are computed through the exact solution of a TDSP-IO on a subgraph $\mathcal{G}^r = (\mathcal{N}^r, \mathcal{A}^r)$ of $\mathcal{G}$, where $\mathcal{N}^r$ and $\mathcal{A}^r$ are restricted to the nodes and arcs of route r, and $\mathcal{N}_c^r$, $\mathcal{N}_e^r$ and $\mathcal{N}_a^r$ are defined similarly. Our model is adapted from that of Koç et al. (2016b). We assume that a driver is paid for on-duty time only, the truck driver has been off-duty and off-the-road for at least 34 consecutive hours before departure from the starting depot (Koç et al., 2016b; Rancourt et al., 2013), each trip has a maximum duration of seven days, and each truck has APU and EPS equipments. The hourly wage of truck drivers is denoted by f^{DRI}, the hourly cost of using an EPS is f^{EPS}, and the hourly cost of using an APU is f^{APU}.

We define the following binary variables. Let $z_i^{rest,EPS}$ be equal to 1 if a rest is taken at EPS location $i \in \mathcal{N}_e^r$, and to 0 otherwise. Let $z_i^{break,EPS}$ be equal to 1 if a break is taken at EPS location $i \in \mathcal{N}_e^r$, and to 0 otherwise. Let $z_i^{rest,IRA}$ be equal to 1 if a rest is taken at IRA location $i \in \mathcal{N}_a^r$, and to 0 otherwise. Let $z_i^{break,IRA}$ be equal to 1 if a break is taken at IRA location $i \in \mathcal{N}_a^r$, and to 0 otherwise. Let $y_i = (y_{i\tau})_{\tau \in T_i}$: $y_{i\tau}$ be equal to 1 if the τ^{th} time window of location $i \in \mathcal{N}^r$ is used, and to 0 otherwise. The following decision variables are defined. Let $u_i^{rest,EPS}$ be the duration of a rest period at EPS location $i \in \mathcal{N}_e^r$. Let $u_i^{break,EPS}$ be the duration of a break period at EPS location $i \in \mathcal{N}_e^r$. Let $u_i^{rest,IRA}$ be the duration of a rest period at IRA location $i \in \mathcal{N}_a^r$. Let $u_i^{break,IRA}$ be the duration of a break period at IRA location $i \in \mathcal{N}_a^r$. Let $x_i^{arrival}$ be the arrival time at location $i \in \mathcal{N}^r$. Let x_i^{start} be the start time of the service, rest or break at location $i \in \mathcal{N}^r$. Let x_i^{end} be the end time of the service, rest or break at location $i \in \mathcal{N}^r$. Let q^{EPS} be the total duration of total EPS idling. Let q^{APU} be the total duration of total APU idling. The mixed integer linear programming

formulation of the TDSP-IO is then:

$$\text{Minimize} \quad f^{DRI}\Bigg(x_{n+1}^{end} - x_0^{start} - \sum_{i\in\mathcal{N}_e^r}\Big(u_i^{rest,EPS} + u_i^{break,EPS}\Big) - \sum_{i\in\mathcal{N}_a^r}\Big(u_i^{rest,IRA} + u_i^{break,IRA}\Big)\Bigg) \tag{4}$$

$$+ f^{EPS}q^{EPS} \tag{5}$$

$$+ f^{APU}q^{APU} \tag{6}$$

subject to

$$x_i^{arrival} = x_i^{start} = x_i^{end} \qquad i \in \mathcal{N}_0 \tag{7}$$

$$x_i^{arrival} \le x_i^{start} \qquad i \in \mathcal{N}_c^r \tag{8}$$

$$x_i^{start} + s_i = x_i^{end} \qquad i \in \mathcal{N}_c^r \tag{9}$$

$$x_i^{arrival} = x_i^{start} \qquad i \in \mathcal{N}_e^r \cup \mathcal{N}_a^r \tag{10}$$

$$x_i^{start} + u_i^{rest,EPS} + u_i^{break,EPS} = x_i^{end} \qquad i \in \mathcal{N}_e^r \tag{11}$$

$$x_i^{start} + u_i^{rest,IRA} + u_i^{break,IRA} = x_i^{end} \qquad i \in \mathcal{N}_a^r \tag{12}$$

$$x_i^{end} + d_{i,i+1} = x_{i+1}^{arrival} \qquad i \in \mathcal{N}^r \tag{13}$$

$$\sum_{\tau\in T_i} y_{i\tau} = 1 \qquad i \in \mathcal{N}^r \tag{14}$$

$$\sum_{\tau\in T_i} y_{i\tau}t_{i\tau}^{min} \le x_i^{start} \qquad i \in \mathcal{N}^r \tag{15}$$

$$x_i^{start} \le \sum_{\tau\in T_i} y_{i\tau}t_{i\tau}^{max} \qquad i \in \mathcal{N}^r \tag{16}$$

$$x_k^{start} - x_i^{end} \le t^{rest} + M\Bigg(\sum_{j=i,j\in\mathcal{N}_e^r}^{k} z_j^{rest,EPS} + \sum_{j=i,j\in\mathcal{N}_a^r}^{k} z_j^{rest,IRA}\Bigg) \qquad i,k \in \mathcal{N}^r, i<k \tag{17}$$

$$x_k^{start} - x_i^{end} \le t^{break} + M\Bigg(\sum_{j=i,j\in\mathcal{N}_e^r}^{k}\Big(z_j^{rest,EPS} + z_j^{break,EPS}\Big) + \sum_{j=i,j\in\mathcal{N}_a^r}^{k}\Big(z_j^{rest,IRA} + z_j^{break,IRA}\Big)\Bigg) \qquad i,k \in \mathcal{N}^r, i<k \tag{18}$$

$$x_{jn+1}^{end} - \sum_{i\in\mathcal{N}_e^r}\Big(u_i^{rest,EPS} + u_i^{break,EPS}\Big) - \sum_{i\in\mathcal{N}_a^r}\Big(u_i^{rest,IRA} + u_i^{break,IRA}\Big) \le t^{duty} \tag{19}$$

$$\sum_{j=i}^{k-1} d_{j,j+1} \le t^{driving} + M\Bigg(\sum_{j=i+1,j\in\mathcal{N}_s^r}^{k-1} z_j^{rest,EPS} + \sum_{j=i+1,j\in\mathcal{N}_a^r}^{k-1} z_j^{rest,IRA}\Bigg) \quad i,k \in \mathcal{N}^r, i<k \tag{20}$$

$$z_i^{rest,EPS} + z_i^{break,EPS} \leq 1 \qquad i \in \mathcal{N}_e^r \qquad (21)$$

$$z_i^{rest,IRA} + z_i^{break,IRA} \leq 1 \qquad i \in \mathcal{N}_a^r \qquad (22)$$

$$u_i^{rest,EPS} \leq M z_i^{rest,EPS} \qquad i \in \mathcal{N}_e^r \qquad (23)$$

$$u_i^{rest,IRA} \leq M z_i^{rest,IRA} \qquad i \in \mathcal{N}_a^r \qquad (24)$$

$$u_i^{break,EPS} \leq M z_i^{break,EPS} \qquad i \in \mathcal{N}_e^r \qquad (25)$$

$$u_i^{break,IRA} \leq M z_i^{break,IRA} \qquad i \in \mathcal{N}_a^r \qquad (26)$$

$$t^{minrest} z_i^{rest,EPS} \leq u_i^{rest,EPS} \qquad i \in \mathcal{N}_e^r \qquad (27)$$

$$t^{minrest} z_i^{rest,IRA} \leq u_i^{rest,IRA} \qquad i \in \mathcal{N}_a^r \qquad (28)$$

$$t^{minbreak} z_i^{break,EPS} \leq u_i^{break,EPS} \qquad i \in \mathcal{N}_e^r \qquad (29)$$

$$t^{minbreak} z_i^{break,IRA} \leq u_i^{break,IRA} \qquad i \in \mathcal{N}_a^r \qquad (30)$$

$$q^{APU} = \sum_{i \in \mathcal{N}_c^r} \left(x_i^{end} - x_i^{arrival} \right) + \sum_{i \in \mathcal{N}_a^r} \left(u_i^{rest,IRA} + u_i^{break,IRA} \right) \qquad (31)$$

$$q^{EPS} = \sum_{i \in \mathcal{N}_e^r} \left(u_i^{rest,EPS} + u_i^{break,EPS} \right) \qquad (32)$$

$$x_i^{arrival} \in [0, t^{horizon}], x_i^{start} \in [0, t^{horizon}], x_i^{end} \in [0, t^{horizon}] \qquad i \in \mathcal{N}^r \qquad (33)$$

$$q^{EPS} \geq 0, q^{APU} \geq 0 \qquad (34)$$

$$z_i^{rest,EPS} \in \{0,1\}, z_i^{break,EPS} \in \{0,1\} \qquad i \in \mathcal{N}_e^r \qquad (35)$$

$$z_i^{rest,IRA} \in \{0,1\}, z_i^{break,IRA} \in \{0,1\} \qquad i \in \mathcal{N}_a^r \qquad (36)$$

$$u_i^{rest,EPS} \geq 0, u_i^{break,EPS} \geq 0 \qquad i \in \mathcal{N}_e^r \qquad (37)$$

$$u_i^{rest,IRA} \geq 0, u_i^{break,IRA} \geq 0 \qquad i \in \mathcal{N}_a^r \qquad (38)$$

$$y_i \in \{0,1\} \qquad i \in \mathcal{N}^r. \qquad (39)$$

The first term (4) of the objective function computes the driver cost during on-duty time. Terms (5) and (6) compute the EPS and APU idling costs, respectively. Constraints (7) guarantee that the departure time from the depot is equal to the start time and to the arrival time. Constraints (8) ensure that the service start time at customer locations is at least equal to the arrival time. Constraints (9) imply that the departure time at customer locations is equal to the sum of the start and service times. Constraints (10) guarantee that the start time at an EPS or at an IRA location is equal to the arrival time. Constraints (11) and (12) state that the departure time at an EPS or at an IRA location is equal to the sum of start, rest and break times. Constraints (13) imply that the arrival time at a location is equal to the end time of the previous location, plus the driving time. Constraints (14) guarantee that exactly one of the time windows is used at each location. Constraints (15) and (16) enforce the time windows restrictions. Constraints (17) and (18) ensure that the time elapsed since the end of the last rest and break period must lie within the regulation parameters, where M is a large number. Constraints (19) imply that the accumulated amount of on-duty hours cannot exceed the seven-day on-duty limit. Constraints (20) guarantee that the total driving hours be-

tween two rest periods does not exceed the daily driving limit $t^{driving}$. Constraints (21) state that at each EPS location at most one rest or break period is scheduled. Constraints (22) impose that at each IRA location at most one rest or break period is scheduled. Constraints (23)–(26) are linking constraints. Constraints (27)–(30) ensure that rest or break periods satisfy the HOS regulations. Constraints (31)–(32) are APU and EPS idling time linking constraints. Finally, constraints (33)–(39) define the domains of the decision variables.

4 Multi-start matheuristic algorithm

Because of the size of the VRTDSP-IO and of the difficulty of computing the C_r values, we solve the problem by means of a multi-start matheuristic algorithm. Matheuristics are often used in vehicle routing (see Archetti and Speranza (2014) for a survey, and Erdoğan et al. (2013) for a recent application). More precisely, the routing part in the SPP is solved heuristically by the ALNS mechanism which generates a set of vehicle routes satisfying (2) and (3), as well as the capacity and t^{duty} constraints imposed by the HOS regulations, without yet considering the time windows. The ALNS framework proposed by Ropke and Pisinger (2006a,b) is based on a removal and insertion iterative procedure. Candidate solutions are accepted or rejected according to a probabilistic simulated annealing criterion. The matheuristic integrates the exact solution of the TDSP-IO (Section 3.2) within the local search. Specifically, every Δ iterations of the algorithm, where Δ is a user-defined parameter, the algorithm identifies a solution having the smallest routing cost in the last Δ iterations. It then computes the total cost of this solution by exactly solving the TDSP-IO for each route r, while satisfying all constraints of the problem, including the time windows.

As in Bräysy et al. (2004) and Palomo-Martínez et al. (2016), we have implemented a multi-start scheme governed by three parameters α, β and γ. The algorithm first generates α initial solutions $\omega_{initial}$ (Algorithm 1), and applies the matheuristic (Algorithm 2) to each of them for $\bar{\kappa} = \beta$ iterations. Taking as $\omega_{initial}$ the best feasible solution ω_{best} thus generated, the matheuristic is then applied for $\bar{\kappa} = \gamma$ iterations. In total, $\alpha\beta + \gamma$ iterations are executed. The case $\alpha = 1$ corresponds to a single-start algorithm.

4.1 Removal and insertion operators

Before proceeding with the description of our matheuristic, we describe the removal and insertion operators it uses. Removal operators remove n' customers and then place them in a removal list $L^{removal}$. The value of n' is selected from the interval $[b_l, b_u]$, where b_l, b_u are input parameters. Insertion operators insert the n' removed customers into a least-cost position of the destroyed solution.

4.1.1 Removal operators

We use the following seven removal operators, the first six having already been used by other authors (see Cherkesly et al., 2015; Koç, 2016; Koç et al., 2016a; Ropke and Pisinger, 2006a,b), and the last one being introduced in this paper.

Random removal (RO1): This operator randomly removes n' customers from a solution.

Worst distance removal (RO2): The aim of this operator is to choose a number of expensive customers based on distance. The cost of a node $j \in \mathcal{N}_c \backslash L^{removal}$ is the sum of its distance from its predecessor i and of its distance to its successor k. The RO2 operator iteratively removes nodes j^* from the solution where $j^* = \arg\max_{j \in \mathcal{N}_c \backslash L^{removal}} \{c_{ij} + c_{jk}\}$.

Shaw removal (RO3): This operator removes a set of n' similar customers. It is iteratively applied to select a node which is most similar to the one last added to $L^{removal}$. The similarity between two customers i and j is defined by a relatedness measure $\Omega(i, j)$ which includes three terms: the distance c_{ij}, an index l_{ij} equal to -1 if i and j are in the same route, and to 1 otherwise, and the demand spread $|q_i - q_j|$. The relatedness measure is given by

$$\Omega(i, j) = \varphi_1 c_{ij} + \varphi_2 l_{ij} + \varphi_3 |q_i - q_j|, \tag{40}$$

where φ_1 to φ_3 are weights that are normalized to find the best candidate solution. The operator starts by randomly selecting a node $i \in \mathcal{N}_c \backslash L^{removal}$, and selects the node j^* to remove, where $j^* = \arg\min_{j \in \mathcal{N}_c \backslash L^{removal}} \{\Omega(i, j)\}$.

Proximity-based removal (RO4): This operator is a special case of RO3 where the selection criterion of a set of customers is solely based on distance.

Demand-based-removal (RO5): This operator is yet another variant of RO3 which is solely based on demand.

Neighborhood removal (RO6): In a given solution with a set of routes, this operator calculates an average distance $\bar{c}(r) = \sum_{(i,j) \in r} c_{ij}/|r|$ for each route r, and selects a node $j^* = \arg\max_{(r \in \mathcal{R}; j \in r)} \{\bar{c}(r) - c_{r \backslash \{j\}}\}$, where $c_{r \backslash \{j\}}$ denotes the average distance of route r, excluding node j.

Pair removal (RO7): This operator, aims to remove pairs of customers which are very close to one another. It first randomly removes $n'/2$ customers, and then removes the other $n'/2$ customers by selecting the nearest ones, based on their distance to the previously removed customers.

4.1.2 Insertion operators

In addition, we introduce three new insertion operators:

Scheduling-greedy insertion (IO1): This operator iteratively inserts all nodes of $L^{removal}$ in the solution, starting with the first customer of $L^{removal}$, by considering the best possible distance based insertion position. For node $i \in \mathcal{N} \backslash L^{removal}$ followed in the destroyed solution by $k \in \mathcal{N} \backslash L^{removal}$, and node $j \in L^{removal}$ we define $\gamma(i, j) = c_{ij} + c_{jk} - c_{ik}$. We find the least-cost insertion position for $j \in L^{removal}$ by $i^* = \arg\min_{i \in \mathcal{N} \backslash L^{removal}} \{\gamma(i, j)\}$. During the insertion process of each customer location, this operator forbids the insertion of a customer in the route if this would violate t^{duty} or the capacity constraints.

Scheduling-greedy insertion with noise function (IO2): This operator is an extension of the IO1 operator that allows a degree of freedom for diversification by selecting the best position for a node. This is done by calculating a noise cost $v(i,j) = \gamma(i,j) + d_{max}\pi\epsilon$, where d_{max} is the maximum distance between all nodes, π is a noise parameter used for diversification, and ϵ is a random number in $[-1, 1]$. We find the least-cost insertion position for $j \in L^{removal}$ by computing $i^* = \arg\min_{i \in \mathcal{N} \setminus L^{removal}} \{v(i,j)\}$. Capacity and t^{duty} constraints are again enforced in the insertion mechanism.

Scheduling-removal list insertion (IO3): This operator is also a variant of the IO1. The outcomes of the IO1 operator are mainly based on the order of the nodes in $L^{removal}$. This operator aims to change this order with the aim of decreasing the insertion costs. The operator first calculates the least insertion costs for each node in $L^{removal}$, then lists them in a non-increasing order, and finally applies IO1.

4.2 Initialization procedure

The initialization procedure presented in Algorithm 1 generates an initial feasible solution $\omega_{initial}$. We first fix the number of removal nodes equal to $|\mathcal{N}_c|$, and we randomly assign all customers to the removal list $L^{removal}$. We then apply the IO2 operator to create solution $\omega^*_{initial}$. We apply CPLEX for each route of $\omega^*_{initial}$. If the solution is feasible, we return it as an initial solution $\omega_{initial}$. Otherwise, we reiterate this procedure until a feasible solution has been identified.

Algorithm 1 Initialization procedure

```
1:  Input: The graph G = (N, A) and related parameters
2:  while a feasible VRTDSP-IO solution is not generated do
3:      Randomly assign all customers to the removal list L^removal
4:      Apply the IO2 operator and create ω*_initial
5:      r̄ ←number of routes of ω*_initial
6:      r ← 1
7:      while r ≤ r̄ do
8:          Apply CPLEX to route r of ω*_initial to solve the TDSP-IO
9:          if route r is feasible then
10:             r ← r + 1
11:         else
12:             Go to line 3
13:     end while
14:     if all routes of ω*_initial are feasible then
15:         ω_initial ← ω*_initial
16: end while
17: Output: ω_initial
```

4.3 Description of the matheuristic

Algorithm 2 presents the general framework of our matheuristic. The iteration counter, the maximum number of iterations, and the frequency of calls to CPLEX are denoted by κ, $\bar{\kappa}$ and Δ,

respectively. It considers partial solutions $\omega^{routing}$ that only specify the vehicle routes, and full solutions ω that also consider scheduling and idling options. We denote by $c(\omega)$ the full cost of ω and by $\underline{c}(\omega^{routing})$ the cost of $\omega^{routing}$, which is only a routing cost. The algorithm uses as input an initial feasible solution $\omega_{initial}$ (line 1). Probabilities are then initialized for each operator (line 2). Initially, they are equal for each removal and insertion operator. The temperature T, used in simulated annealing, is initially set at $c(\omega_{initial})T_0$, and T_0 is the startup temperature defined as an input parameter. Initially, the promising, current and best solutions coincide with the first solution. These are denoted by $\omega_{promising}$, $\omega_{current}$, and ω_{best}, respectively. A temporary solution is denoted by $\omega_{temporary}$. Such a solution will either be discarded or become the current solution.

In lines (5–15), the driver and idling costs, and the feasibility of the routing in terms of time windows and HOS regulations are neglected. A removal operator is selected, applied to the $\omega_{current}$, and a destroyed temporary solution $\omega_{temporary}*$ is obtained at the beginning of every iteration (line 5). An insertion operator is then applied to $\omega_{temporary}*$ to yield $\omega_{temporary}$ (line 6). If the routing cost of $\omega_{temporary}$ is less than that of $\omega_{promising}$, it is then replaced with $\omega_{temporary}$ (lines 7 and 8). If the routing cost of $\omega_{temporary}$ is less than that of $\omega_{current}$, it is then replaced with $\omega_{temporary}$ (lines 9 and 10). Otherwise (lines 11 and 15), the probability ϑ of accepting a non-improving solution is computed as a function of the current temperature. A random number ϵ is generated in the interval $[0, 1]$; if ϵ is less than ϑ, $\omega_{current}$ is then replaced by $\omega_{temporary}$. If κ is a multiple of Δ (line 16–21), $\omega_{current}$ is replaced by $\omega_{promising}$ and the TDSP-IO is solved by CPLEX for each route of $\omega_{current}$. If this schedule is feasible and the total cost of $\omega_{current}$ is less than that of ω_{best}, then ω_{best} is replaced by $\omega_{current}$. The temperature is gradually decreased during the algorithm as δT (line 22), where $0 < \delta < 1$ is a fixed cooling parameter.

An adaptive weight adjustment procedure (lines 23 and 24) is then applied to update the probabilities of the removal and insertion operators. Each removal and insertion operator has a certain probability of being chosen at every iteration. The selection criterion is based on the historical performance of the operators and is calibrated by a roulette-wheel mechanism as in Ropke and Pisinger (2006a,b). After every segment of ρ iterations, the probability of each operator is recalculated according to its total score. Initially, the probabilities of each removal and insertion operator are equal. Let ϕ_i^t be the probability of selecting operator i in segment t, and let $\phi_i^{t+1} = \phi_i^t(0.9) + 0.1\Pi_i/\zeta_i$, for operator i; Π_i is the score and ζ_i is the number of times it was used during the last segment. At the start of each segment, the scores of all operators are set to zero. The scores are increased by σ_1 if a new best solution is found, by σ_2 if the new solution is better than the current solution, and by σ_3 if the new solution is worse than the current solution.

Finally, algorithm returns the best found feasible solution ω_{best}, that is, a set of routes r together with an optimal truck driver schedule and idling options for each of them.

Algorithm 2 General framework of the matheuristic

Input: An initial solution $\omega_{initial}$, a maximum number of $\bar{\kappa}$ iterations, and a segment length ρ
Initialize the probabilities associated with the operators
$T \leftarrow c(\omega_{initial})T_0$, $\omega_{promising} \leftarrow \omega_{current} \leftarrow \omega_{best} \leftarrow \omega_{initial}$, $\kappa \leftarrow 1$
while $\kappa \leq \bar{\kappa}$ **do**
    Select a removal operator, apply it to the $\omega_{current}$ and obtain $\omega_{temporary}*$
    Select an insertion operator, apply it to the $\omega_{temporary}*$ and obtain $\omega_{temporary}$
    if $\underline{c}(\omega^{routing}_{temporary}) < \underline{c}(\omega^{routing}_{promising})$ **then**
        $\omega_{promising} \leftarrow \omega_{temporary}$
    if $\underline{c}(\omega^{routing}_{temporary}) < \underline{c}(\omega^{routing}_{current})$ **then**
        $\omega_{current} \leftarrow \omega_{temporary}$
    else
        $\vartheta \leftarrow e^{-(\underline{c}(\omega^{routing}_{temporary}) - \underline{c}(\omega^{routing}_{current}))/T}$
        Generate a random number ϵ
        if $\epsilon < \vartheta$ **then**
            $\omega_{current} \leftarrow \omega_{temporary}$
    if κ is a multiple of Δ **then**
        $\omega_{current} \leftarrow \omega_{promising}$
        Solve the TDSP-IO on $\omega_{current}$ by CPLEX
        if the schedule is feasible **then**
            if $c(\omega_{current}) < c(\omega_{best})$ **then**
                $\omega_{best} \leftarrow \omega_{current}$
    $T \leftarrow \delta T$
    if κ is a multiple of ρ **then**
        Update the probabilities of the operators
    $\kappa \leftarrow \kappa + 1$
end while
Output: The best feasible solution ω_{best}

5 Computational experiments and analyses

This section present the results of our computational experiments. All computations were performed on an Intel 3.6 GHz processor and 32 GB of RAM. The matheuristic was implemented in C++. We used CPLEX 12.6 with its default settings as the optimizer to solve the TDSP-IO. In all experiments, five runs were performed on each instance, as in Rancourt et al. (2013) and Goel and Vidal (2014), and the best one is reported.

5.1 Benchmark instances

We generated the benchmark instances for the VRTDSP-IO by considering the data set described by Goel (2009) and Rancourt et al. (2013) for the VRTDSP which are derived from the classical Solomon (1987) VRPTW instances with 100 nodes. This set includes 56 instances, consisting of a clustered data set C, a random data set R, and a semi-clustered data set RC. The sets C1, R1 and RC1 have a short scheduling horizon and small vehicle capacities, in contrast to the sets C2, R2 and RC2 which have a longer scheduling horizon and larger vehicle capacities. The time windows and the travel time matrix adapted for the VRTDSP context. The service time of each customer is set to one hour. Geographic coordinates of nodes, customer demands and vehicle capacities remain the same. We considered a seven-day horizon ($t^{horizon} = 168$ h). The time intervals in the original Solomon instances were defined over periods of 24 hours. Here we replicate these time windows for each of the seven days. Instead of considering vehicle speed as 60 distance units per h as in the original Solomon instances, the speed is set to five distance units per h. To represent the real road network, we assume that one distance unit is 15 km. Thus, the speed is equal to 75 km/h. Distances are rounded to a single decimal place.

We assumed that an IRA is located every 100 km on every route and that an EPS is located every 340 km on every route, as in Koç et al. (2016b). For simplicity, we assume that the fuel cost is \$0.66/L (\$2.50/US gallon) (EIA, 2016), the routing cost is \$0.10/km, $f^{EPS} = \$1.00$/h (Argonne National Laboratory, 2015), $f^{APU} = \$0.63$/h (0.95 L/h x \$0.66/L) (Argonne National Laboratory, 2015; EIA, 2016), $f^{DRI} = \$17.34$/h (Pay Scale, 2016).

5.2 Parameter calibration on training instances

To calibrate the parameters, we initially worked with $\alpha = 1$, $\beta = 100$ and $\gamma = 9900$, i.e., in a single-start fashion. We applied a calibration procedure on six representative 100-customer training instances, C101, C203, R101, R211, RC105 and RC207. The parameter values selected after the calibration phase are provided in Table 2.

5.3 Sensitivity analyses

We have conducted extensive sensitivity experiments to comparatively analyze several variations of the matheuristic.

Table 2: Parameters used in the matheuristic

Description	Value
Total number of iterations ($\alpha\beta + \gamma$)	10000
Frequency of calls to CPLEX (Δ)	200
Roulette wheel parameter (ρ)	450
New global solution score σ_1	1
Better solution score σ_2	0
Worse solution score σ_3	5
Startup temperature parameter (T_0)	100
Cooling parameter (δ)	0.999
Lower limit of removable nodes (b_l)	10% of $\lvert\mathcal{N}_c\rvert$
Upper limit of removable nodes (b_u)	30% of $\lvert\mathcal{N}_c\rvert$
First Shaw parameter φ_1	0.5
Second Shaw parameter φ_2	0.15
Third Shaw parameter φ_3	0.25
Noise parameter π	0.1

5.3.1 Effect of the score parameters

We investigated the impact of new global solution (σ_1), better solution (σ_2) and worse solution (σ_3) parameters on the solution quality. To this end, we analyzed four combinations of σ_1, σ_2 and σ_3. The results are presented in Table 3. The column Dev (%) shows the percentage deviation from the best-known solution. The best setting for all instances is obtained with $\sigma_1 = 1, \sigma_2 = 0$ and $\sigma_3 = 5$.

Table 3: Performance of the matheuristic for various σ_1, σ_2 and σ_3 settings.

$(\sigma_1, \sigma_2, \sigma_3)$	(5,1,0)	(3,1,0)	(1,0,3)	(1,0,5)
Instance	Dev (%)	Dev (%)	Dev (%)	Dev (%)
C101	0.41	0.32	0.12	0.00
C203	0.31	0.24	0.22	0.00
R101	0.25	0.22	0.10	0.00
R211	0.33	0.37	0.28	0.00
RC105	0.41	0.38	0.37	0.00
RC207	0.45	0.32	0.21	0.00
Average	0.36	0.31	0.22	0.00

5.3.2 Effect of the parameter Δ which controls the frequency of calls to CPLEX

We analyzed the impact on solution quality and computing time of the parameter Δ which regulates the frequency of calls to CPLEX to solve the TDSP-IO. Table 4 shows the average values on the training instances. The column CPU (min) shows the total computation time of five runs in minutes. It can be seen that the best setting in terms of solution quality is obtained for $\Delta = 200$.

5.3.3 Effect of the multi-start scheme

Here we investigate the effect on solution quality of applying a multi-start scheme with parameters (α, β, γ). We tested different versions of it and compared the results against the single-start base case $(\alpha, \beta, \gamma) = (1, 100, 9900)$ and $\Delta = 200$. These versions are $(\alpha, \beta, \gamma) = (1, 100, 4900)$, $(5, 100, 4500)$, $(12, 100, 3800)$, $(25, 100, 2500)$, $(37, 100, 1300)$, $(10, 100, 9000)$, $(25, 100, 7500)$, $(50, 100, 5000)$,

Table 4: Performance of the matheuristic for various Δ settings.

Instance	$\Delta = 100$		$\Delta = 200$		$\Delta = 300$		$\Delta = 400$		$\Delta = 500$	
	Dev (%)	CPU (min)	Dev (%)	CPU (min)	Dev (%)	CPU (min)	Dev (%)	CPU (min)	Dev (%)	CPU (min)
C101	0.09	85.2	0.00	76.1	0.13	70.5	0.22	64.3	0.26	54.1
C203	0.14	95.6	0.00	85.7	0.12	81.0	0.23	65.7	0.28	59.8
R101	0.12	95.4	0.00	91.2	0.18	84.0	0.19	74.8	0.32	58.7
R211	0.08	89.1	0.00	75.3	0.13	71.3	0.24	60.7	0.38	55.4
RC105	0.10	69.3	0.00	61.2	0.18	58.0	0.19	50.8	0.27	44.2
RC207	0.09	104.2	0.00	101.1	0.12	72.0	0.17	64.0	0.25	54.3
Average	0.10	89.8	0.00	81.8	0.14	72.8	0.21	63.4	0.29	54.4

and $(75, 100, 2500)$. The first five versions ran for 5,000 iterations while the others ran for 10,000 iterations. Table 5 provides the average deviations with respect to the base case. The results clearly show that most of these versions improve upon the base case. The two versions $(\alpha, \beta, \gamma) = (25, 100, 2500)$ and $(37, 100, 1300)$ yield better results than $(1, 100, 9900)$, which requires 10,000 iterations. Our results indicate that multi-start scheme yields better results than the single-start scheme. For the rest of the experiments, we work with $(\alpha, \beta, \gamma) = (25, 100, 7500)$ which becomes the updated base case.

Table 5: Average deviations of the effect of multi-start procedure with respect to the base case.

(α, β, γ)	Dev (%)	(α, β, γ)	Dev (%)
(1,100,4900)	1.79	(1,100,9900)	0.00
(5,100,4500)	0.56	(10,100,9000)	-0.99
(12,100,3800)	0.32	(25,100,7500)	-2.13
(25,100,2500)	-1.12	(50,100,5000)	-1.22
(37,100,1300)	-1.25	(75,100,2500)	-1.36

5.3.4 Marginal impact of the operators

We now analyze the marginal value of the operators. To this end, we removed one operator at a time by keeping all the remaining ones. Table 6 shows the average percentage deviations of the solutions obtained by removing each operator individually with respect to the results yielded by the updated base case. Our results show that if we discard any operator, the solution worsens. This indicates that all of the operators have positive marginal impact on solution quality. Therefore, we decided to keep all of them.

Table 6: Average deviations of the effect of the removal of operators with respect to the base case.

Removed operator	Dev (%)	Removed operator	Dev (%)	Removed operator	Dev (%)
RO1	0.07	RO5	0.08	IO1	0.22
RO2	0.15	RO6	0.09	IO2	0.32
RO3	0.14	RO7	0.12	IO3	0.25
RO4	0.08				

5.4 Results on the benchmark instances

Table 7 presents the average results obtained on benchmark instances of the VRTDSP-IO, excluding the six training instances. The first column shows the instance type where the number in parentheses represents the total number of instances. The other columns display the routing cost ($), the driver cost ($), the EPS idling cost ($), the APU idling cost ($), the total cost ($), and the total CPU time in minutes over five runs. For detailed results of all instances, the reader is referred to Table A.1 in the Appendix which reports the results corresponding to the best solution for each instance. One can see that all CPU times are below two hours, which is reasonable for a problem that is typically solved on a weekly basis.

Table 7: Average results on the VRTDSP-IO instances except six training instances.

Instance set	Routing cost ($)	Driver cost ($)	EPS cost ($)	APU cost ($)	Total cost ($)	CPU (min)
C1 (8)	2815.0	8488.3	5.1	505.7	11814.2	81.4
C2 (7)	3551.5	10015.0	1.6	654.0	14222.1	78.5
R1 (11)	2449.9	7446.8	1.3	428.4	10326.5	92.4
R2 (10)	2321.5	7198.8	5.1	431.7	9957.1	76.5
RC1 (7)	2837.8	8412.1	1.5	477.6	11729.0	60.5
RC2 (7)	3117.4	9093.0	4.0	496.6	12711.0	104.3

6 Conclusions

We have introduced the Vehicle Routing and Truck Driver Scheduling Problem with Idling Options (VRTDSP-IO) using a comprehensive objective function that minimizes routing costs, driver costs, as well as Electrified Parking Space and Auxiliary Power Unit usage costs. We have developed a matheuristic capable of efficiently tackling the problem. It solves a set partitioning model heuristically by generating good vehicle routes through a multi-start adaptive large neighborhood search mechanism. Promising vehicle routes are optimized by periodically solving exactly by CPLEX a truck driver scheduling problem with idling options. One interesting feature of our algorithm is the use of a multi-start scheme which has proved highly beneficial in terms of solution quality. Our algorithm was tested on instances derived from the Solomon test-bed, and several sensitivity analyses were conducted. We have shown that the proposed algorithm can efficiently solve the VRTDSP-IO for the first time within a reasonable computing effort.

Acknowledgements

The authors gratefully acknowledge funding provided by the Canadian Natural Sciences and Engineering Research Council under grants 2015-06189 and 436014-2013.

Appendix

Table A.1 presents the detailed results on all benchmark instances for the VRTDSP-IO.

Table A.1: Detailed results on the VRTDSP-IO instances.

Instance	Routing cost ($)	Driver cost ($)	EPS cost ($)	APU cost ($)	Total cost ($)	CPU (min)
C101	3903.7	10788.9	33.5	685.2	15411.3	82.5
C102	3251.2	9405.2	0.0	604.7	13261.1	81.0
C103	2317.5	7369.5	0.5	471.1	10158.6	79.0
C104	1858.5	6030.8	0.0	331.9	8221.2	72.5
C105	3321.0	10149.1	10.0	624.8	14104.9	79.0
C106	3728.2	10579.1	10.0	664.5	14981.8	79.0
C107	3136.5	9471.1	10.0	492.4	13110.0	86.0
C108	2730.0	8045.7	10.0	452.1	11237.8	83.5
C109	2177.2	6856.2	0.0	404.4	9437.8	91.5
C201	4851.7	13228.6	0.0	963.2	19043.5	72.5
C202	4178.2	11421.8	0.5	767.0	16367.5	77.5
C203	3126.7	8963.0	0.5	559.2	12649.4	74.0
C204	2071.5	6523.3	0.0	340.3	8935.0	82.5
C205	4060.5	11194.7	10.0	717.2	15982.4	69.5
C206	3405.0	9606.3	0.5	652.5	13664.3	71.0
C207	3194.2	9119.1	0.0	589.8	12903.0	84.0
C208	3099.7	9011.5	0.0	547.9	12659.1	92.5
R101	4488.7	12274.9	0.5	821.1	17585.1	87.0
R102	3667.5	10247.9	10.5	611.8	14537.7	93.5
R103	2667.7	7967.7	1.5	508.4	11145.3	85.5
R104	1878.0	6075.9	1.0	384.1	8339.0	92.5
R105	3590.2	10072.8	0.9	543.0	14206.9	86.0
R106	2850.7	8607.5	0.0	538.3	11996.5	97.0
R107	2442.7	7381.6	0.0	412.0	10236.3	98.5
R108	1755.7	5793.2	0.0	302.3	7851.2	92.5
R109	2322.0	7102.4	0.5	437.4	9862.3	87.0
R110	2164.5	6776.4	0.0	358.5	9299.4	92.0
R111	2108.2	6608.2	0.0	372.3	9088.6	98.0
R112	1501.5	5281.7	0.0	244.7	7027.9	94.0
R201	3378.0	9791.8	10.0	603.7	13783.5	70.5
R202	2960.2	8591.9	10.0	552.8	12114.8	73.5
R203	2398.5	7490.8	10.4	416.2	10315.9	72.5
R204	1674.7	5609.4	0.5	278.6	7563.2	74.5
R205	2556.0	7700.6	0.0	487.8	10744.4	65.5
R206	2132.2	6861.4	20.0	431.4	9444.9	76.5
R207	1797.7	5890.3	0.0	340.7	8028.7	80.5
R208	1410.7	5117.0	0.0	227.5	6755.2	93.0
R209	2227.5	7005.3	0.0	522.5	9755.3	71.0
R210	2679.7	7929.5	0.5	455.8	11065.5	87.0
R211	1631.2	5505.4	14.3	293.3	7444.2	66.5
RC101	4278.0	11831.0	1.0	695.8	16805.8	57.0
RC102	3507.7	10045.0	0.0	703.0	14255.7	63.0
RC103	2420.2	7357.3	9.1	448.2	10234.8	69.5
RC104	2043.0	6457.4	0.0	358.4	8858.8	58.0
RC105	4405.5	12032.2	0.0	629.8	17067.5	61.5
RC106	2958.0	8758.4	0.0	460.5	12176.9	56.0
RC107	2467.5	7561.9	0.5	358.9	10388.8	60.0
RC108	2190.0	6873.5	0.0	318.6	9382.1	60.0
RC201	4760.2	13067.4	17.4	647.0	18492.0	111.5
RC202	3814.5	10591.2	0.5	634.2	15040.4	104.5
RC203	2325.0	7109.3	0.0	415.4	9849.6	99.0
RC204	1750.5	5781.1	10.0	282.7	7824.3	101.5
RC205	3870.7	10908.5	0.0	637.6	15416.8	101.0
RC206	3329.2	9901.1	0.0	551.0	13781.3	109.0
RC207	2621.2	7796.0	0.0	435.8	10853.0	106.5
RC208	1971.7	6292.6	0.0	308.0	8572.3	103.5

References

Archetti, C., Savelsbergh, M. W. P., 2009. The trip scheduling problem. Transportation Science 43, 417–431.

Archetti, C., Speranza, M. G., 2014. A survey on matheuristics for routing problems. EURO Journal on Computational Optimization 2, 223–246.

Argonne National Laboratory, 2015. Long-haul truck idling burns up profits. United States Department of Energy Argonne National Laboratory. <http://www.anl.gov/sites/anl.gov/files/es_long-haul_truck_idling_factsheet_Sept2015.pdf> (accessed 25.07.2016).

Bräysy, O., Hasle, G., Dullaert, W., 2004. A multi-start local search algorithm for the vehicle routing problem with time windows. European Journal of Operational Research 159, 586–605.

Carrier, 2016. Auxiliary Power Unit. http://www.carrier.com/truck-trailer/en/north-america/products/na-truck-trailer/special-products/auxiliary-power-unit/> (accessed 03.01.2016).

Cherkesly, M., Desaulniers, G., Laporte, G., 2015. A population-based metaheuristic for the pickup and delivery problem with time windows and LIFO loading. Computers & Operations Research 62, 23–35.

DOE, 2016a. Truck stop electrification locator. United States Department of Energy Alternative s Data Center. <http://www.afdc.energy.gov/tse_locator/> (accessed 08.01.2016).

EIA, 2016. United States on-highway diesel fuel prices, United States Energy Information Administration. <https://www.eia.gov/petroleum/gasdiesel/> (accessed 15.07.2016).

EnviroDock, 2016. <http://www.envirodock.com/> (accessed 14.01.2016).

Erdoğan, G., McLeod, F., Cherrett, T., Bektaş, T., 2013. Matheuristics for solving a multi-attribute collection problem for a charity organisation. Journal of the Operational Research Society 66, 177–190.

FMCSA, 2014. Hours of service. United States Federal Motor Carrier Safety Administration. <https://www.fmcsa.dot.gov/regulations/hours-of-service> (accessed 07.12.2015).

Goel, A., 2009. Vehicle scheduling and routing with drivers working hours. Transportation Science 43, 17–26.

Goel, A., 2012a. The minimum duration truck driver scheduling problem. EURO Journal on Transportation and Logistics 1, 285–306.

Goel, A., 2012b. The Canadian minimum duration truck driver scheduling problem. Computers & Operations Research 39, 2359–2367.

Goel, A., 2012c. A mixed integer programming formulation and effective cuts for minimising schedule durations of Australian truck drivers. Journal of Scheduling 15, 733–741.

Goel, A., Archetti, A., Savelsbergh, M. W. P., 2012. Truck driver scheduling in Australia. Computers & Operations Research 39, 1122–1132.

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

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

Goel, A., Irnich, S., 2016. An exact method for vehicle routing and truck driver scheduling problems. Transportation Science, in press.

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

Koç, Ç., 2016. A unified-adaptive large neighborhood search metaheuristic for periodic location-routing problems. Transportation Research Part C 68, 265–284.

Koç, Ç., Bektaş, T., Jabali, O., Laporte, G., 2015. A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems. Computers & Operations Research 64, 11–27.

Koç, Ç., Bektaş, T., Jabali, O., Laporte, G., 2016a. The fleet size and mix location-routing problem with time

windows: Formulations and a heuristic algorithm. European Journal of Operational Research 248, 33–51.

Koç, Ç., Bektaş, T., Jabali, O., Laporte, G., 2016b. A comparison of three idling options in long-haul truck scheduling. Transportation Research Part B 93, 631–647.

Kok, A. L., Meyer, C. M., Kopfer, H., Schutten, J. M. J., 2010. A dynamic programming heuristic for the vehicle routing problem with time windows and European community social legislation. Transportation Science 44, 442–454.

Lahyani, R., Khemakhem, M., Semet, F., 2015. Rich vehicle routing problems: From a taxonomy to a definition. European Journal of Operational Research 241, 1–14.

Laporte, G., 2016. Scheduling issues in vehicle routing. Annals of Operations Research 236, 463–474.

McCartt, A.T., Hellinga, L.A., Solomon, M.G., 2008. Work schedules of long-distance truck drivers before and after 2004 hours-of service rule change. Traffic Injury Prevention 9, 201–210.

NREL, 2016. The National Renewable Energy Laboratory. <http://www.nrel.gov/> (accessed 06.01.2016).

Palomo-Martínez, P. J., Salazar-Aguilar, M. A., Laporte, G., 2016. Planning a selective delivery schedule through adaptive large neighborhood search. CIRRELT working paper, Montréal.

Pay Scale, 2016. United States truck driver salary. Pay Scale Inc. <http://www.payscale.com/research/US/Job=Truck_Driver%2c_Heavy_%2f_Tractor-Trailer/Hourly_Rate> (accessed 06.01.2016).

Prescott-Gagnon, E., Drexl, M., Rousseau, L.-M., 2010. European driver rules in vehicle routing with time windows. Transportation Science 44, 455–473.

Rahman, S. A., Masjuki, H. H., Kalam, M. A., Abedin, M. J., Sanjid, A., Sajjad, H., 2013. Impact of idling on fuel consumption and exhaust emissions and available idle-reduction technologies for diesel vehicles–A review. Energy Conversion and Management 74, 171–182.

Rancourt, M.-È., Cordeau, J.-F., Laporte, G., 2013. Long-haul vehicle routing and scheduling with working hour rules. Transportation Science 47, 81–107.

Ropke, S., Pisinger, D., 2006a. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science 40, 455–472.

Ropke, S., Pisinger, D., 2006b. A unified heuristic for a large class of vehicle routing problems with backhauls. European Journal of Operational Research 171, 750–775.

Shorepower Technologies, 2016. <http://www.shorepower.com/> (accessed 05.01.2016).

Solomon, M. M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 35, 254–265.

Toth, P. Vigo, D., eds. 2014. Vehicle Routing: Problems, Methods, and Applications. MOS-SIAM Series on Optimization, Philadelphia.

Tunalıoğlu, R., Koç, Ç., Bektaş, T., 2016. A multiperiod location-routing problem arising in the collection and treatment of olive oil mill wastewater, Journal of the Operational Research Society 67, 1012–1024.

US Rest Areas, 2016. United States Interstate Rest Areas. <http://restareas.appspot.com/> (accessed 05.08.2016).

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