



Production, Manufacturing and Logistics

Optimizing departure times in vehicle routes

A.L. Kok^{a,*}, E.W. Hans^b, J.M.J. Schutten^b^a Algorithmic R&D, ORTEC, P.O. Box 490, 2800AL Gouda, The Netherlands^b Operational Methods for Production and Logistics, University of Twente, P.O. Box 217, 7500AE Enschede, The Netherlands

ARTICLE INFO

Article history:

Received 11 August 2009

Accepted 10 October 2010

Available online 30 October 2010

Keywords:

Integer programming

Departure time scheduling

Time-dependent travel times

Driving hours regulations

ABSTRACT

Most solution methods for the vehicle routing problem with time windows (VRPTW) develop routes from the earliest feasible departure time. In practice, however, temporary traffic congestion make such solutions non-optimal with respect to minimizing the total duty time. Furthermore, the VRPTW does not account for driving hours regulations, which restrict the available travel time for truck drivers. To deal with these problems, we consider the vehicle departure time optimization (VDO) problem as a post-processing of a VRPTW. We propose an ILP formulation that minimizes the total duty time. The results of a case study indicate that duty time reductions of 15% can be achieved. Furthermore, computational experiments on VRPTW benchmarks indicate that ignoring traffic congestion or driving hours regulations leads to practically infeasible solutions. Therefore, new vehicle routing methods should be developed that account for these common restrictions. We propose an integrated approach based on classical insertion heuristics.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

The VRP, which concerns the scheduling and routing of a homogeneous vehicle fleet among a set of customers, has been widely discussed in the literature (Toth and Vigo (2002) present an extensive overview of the VRP and solution methods). The problem arises in many application areas such as retail distribution, mail delivery, freight operations, school bus routing, and dial-a-ride service. However, two real-life restrictions have hardly been discussed: temporary traffic congestion and driving hours regulations. This paper addresses a variant of the vehicle routing problem with time windows (VRPTW) in which these real-life conditions are incorporated.

Traffic congestion forms a major problem for businesses such as logistic service providers and distribution firms. Due to temporary traffic congestion, vehicles arrive late at customers and driving hours regulations are violated. Since travel times depend on both distance traveled and time of departure, Malandraki and Daskin (1992) introduce the time dependent vehicle routing problem (TDVRP). Furthermore, Hill and Benton (1992), Ichoua et al. (2003), Fleischmann et al. (2004), Haghani and Jung (2005), and Van Woensel et al. (2008) propose travel time models and algorithms for the TDVRP.

Driving hours regulations severely restrict the set of feasible vehicle routes in a VRP. These regulations impose restrictions on

the total daily travel time available for a truck driver, as well as requirements on the scheduling of (lunch-) breaks during the day. Xu et al. (2003) consider a practical pickup and delivery problem in which the US hours of service regulations are considered. They conjecture that finding a feasible driver schedule after the vehicle routes are constructed is an NP-hard problem. However, Archetti and Savelsbergh (2009) develop a polynomial time algorithm for this problem that runs in cubic time. Goel and Kok (2009b) propose an improved algorithm for this problem that runs in quadratic time and Goel and Kok (2009a) adapt this algorithm for the European Legislation on driving hours for team truck drivers. Goel (2009) considers the VRPTW with the European Legislation on driving and working hours. He proposes a labeling algorithm for determining the feasibility of vehicle routes with respect to these regulations and embeds this algorithm in a large neighborhood search algorithm. Kok et al. (2010) propose a restricted dynamic programming heuristic for this problem that substantially improves the results found by Goel (2009) in substantially smaller computation times. However, neither of the mentioned papers considers time-dependent travel times.

Since travel times in practice depend on the times of departure, and the amount of driving and duty time available to a truck driver is limited by driving hours regulations, the feasibility of a route depends on the chosen departure times. Furthermore, the costs of a truck driver depend on the total time the truck driver is on duty, i.e., the difference between his departure time and return time at the depot. Therefore, it is profitable to minimize a truck driver's duty time by departure time optimization. Minimizing the duty times also minimizes the total time a vehicle is in use, which is

* Corresponding author. Tel.: +31 182 540 500; fax: +31 182 540 540.

E-mail addresses: leendert.kok@ortec.com (A.L. Kok), e.w.hans@utwente.nl (E.W. Hans), m.schutten@utwente.nl (J.M.J. Schutten).

of high value for logistic service providers and distribution firms. For an extensive overview of (multi-) objective functions within vehicle routing, we refer to Jozefowicz et al. (2008). The only papers that we are aware of that consider minimizing route duration in the objective for the VRPTW are of Savelsbergh (1992), Sessomboon et al. (1998), Hong and Park (1999), Geiger (2001), and Baràn and Schaefer (2003).

Since more information on historical travel speeds during each time of day is available, time-dependent travel times can now be better estimated. This information is already used by several route-planners on the Internet to provide travel time estimations depending on travel date and time of day to individual drivers. An example is the on-line route planner of the Dutch motorists' organization ANWB. This route planner provides travel time estimations based on historical information on time and location dependent travel speeds using a travel time estimator developed by the Dutch company TNO. Another example that demonstrates the positive impact of using historical travel time data to construct vehicle routes off-line is of Eglese et al. (2006). For their analysis, they use a so-called Road Timetable™ produced by the UK road networking system ITIS Floating Vehicle Data. This Road Timetable™ contains information on time-dependent travel times for a road network based on a record of past road conditions so that travel times can be related to time of the day, day of the week, and season of the year. Since time-dependent travel times can now be better estimated, we consider deterministic traffic congestion in this paper.

On top of these new opportunities for high quality off-line travel time estimations, compact duty times in off-line vehicle route plans have a strong positive impact on the overall quality of vehicle routing solutions. This point was stressed by the Dutch company ORTEC (Gromicho, 2008), a key-player in the vehicle routing systems market. Therefore, optimizing departure times off-line is highly profitable in practice.

To the best of our knowledge, this is the first paper which addresses the vehicle departure time optimization problem (VDO), with time-dependent travel times. We first approach the VDO as a post-processing step of solving a VRPTW and propose an ILP formulation for it. Next, we propose a construction heuristic, based on this ILP formulation, as a first integrated approach for the VRPTW with time-dependent travel times and driving hours regulations. There are two main reasons for this approach.

First, a solution method for the VDO as post-processing can be directly applied in practice. As ORTEC indicated, in practice departure times are optimized after the vehicle routes have been constructed (by routing software or by hand).

Second, it is computationally expensive to incorporate departure time optimization within sophisticated solution methods for the VRP. A change of departure time (caused by, e.g., inter-route customer swap or customer insertion in a route) at one customer results in different departure times at its succeeding customers. Therefore, the costs and feasibility of such changes cannot be calculated in constant time, but requires at least linear time (e.g., when continuing ASAP from the inserted customer). Since driving and duty times are restricted by driving hours regulations, it may turn out that the route is only feasible if the departure time at some customer is delayed, such that the total driving time reduces, while the total duty time increases. Therefore, determining the costs and feasibility of, e.g., a customer insertion, may require more than linear time. Furthermore, we are not aware of any paper that addresses the complex problem of both scheduling and routing vehicles under time-dependent travel times and driving hours regulations. Next, departure time optimization, which has only been applied to models without time-dependent travel times or driving hours regulations, is much harder under these real-life restrictions.

The contributions of this paper are the following. First, it proposes an exact solution method for the VDO as a post-processing step, which is valuable for practice. This practical value is demonstrated by a case study in which departure time optimization reduces duty times by 15% on average. Second, computational experiments on VRPTW benchmarks indicate that vehicle routing models that do not account for either time-dependent travel times or driving hours regulations are in general not feasible in practice. Therefore, this paper clearly shows the need for the development of algorithms that build vehicle routes that incorporate both time-dependent travel times and driving hours regulations. Third, it proposes a first integrated approach for solving the VRPTW with time-dependent travel times and driving hours regulations.

This paper is organized as follows. Section 2 formally introduces the VDO. Next, Section 3 proposes an ILP formulation for the VDO and discusses the modeling of the time-dependent travel times in the ILP formulation. We test the ILP formulation in Section 4 on problem instances of realistic sizes, and we propose a first integrated solution method for the VRPTW with time-dependent travel times and driving hours regulations in Section 5. Section 6 shows that our approach is flexible with respect to several practical extensions and Section 7 concludes the paper.

2. Problem description VDO

We first approach the VDO as a post-processing step of the VRPTW. If the VDO turns out to be infeasible, then the vehicle routes constructed in the first phase should be modified. This may be very costly, since more vehicles and more duty time are needed to serve all customers within their time windows and respecting the driving hours regulations. In Section 4, we discuss ways to avoid such costly route modifications. The input of the VDO is an ordered set of customers $i = 0, \dots, n + 1$, which need to be served in this order. We consider a deterministic planning problem, in which travel times are also considered to be deterministic (but time-dependent). For simplicity reasons, we first assume that all customers have to be served on one day. In addition, since in practice breaks are usually scheduled at customers, we first assume that breaks can only be taken at customers. There are exceptions, especially in long distance (international) transports where breaks are also scheduled at parking lots along the routes. We show in Section 6 how our ILP formulation can be extended to the case where breaks can also be scheduled at parking lots, and we show how to extend our ILP formulation to multi-day planning.

Each customer i has given a time window $[e_i, l_i]$ in which its service has to start. The service time of each customer is given by s_i . The travel time between two successive customers i and $i + 1$ is given by $c_i(X_i^d)$, where X_i^d is the chosen departure time from customer i . The chosen departure times at the customers are restricted by driving hours regulations.

Since driving hours regulations are country dependent, it might be hard to propose a general formulation covering the driving hours regulations of each country in the world. Since the European driving hours regulations (European Union, 2006) are more restrictive than the North-American ones (Federal Motor Carrier Safety Administration, 2008) and they are valid for all member countries of the European Union, we base our formulation on the European driving hours regulations. These regulations consist of four components:

1. A truck driver is not allowed to drive more than 9 hours (t_{max}) on a day.
2. A period between two breaks of at least 0.75 hours (b_{total}) is called a *driving period*. The accumulated driving time in a driving period may not exceed 4.5 hours (t_{dr}). The break that

ends a driving period may be reduced to 0.5 hours (b_{min}^1) if an additional break of at least 0.25 hours (b_{min}^2) is taken anywhere during that driving period. We call a break of at least b_{min}^1 (b_{min}^2) hours a break of type 1 (2). Therefore, each type 1 break is also a type 2 break.

3. The driving hours regulations do not allow service time at customers to be considered as break time. Therefore, if a truck driver takes a break at a customer, he can do that before or after serving the customer, or both. However, each waiting period before and after serving a customer should be checked separately whether it can be considered a break of type 1 or 2.
4. A truck driver is not allowed to be on duty for more than 13 hours (d_{max}).

These regulations apply throughout the entire European Union and they are hard constraints. There are some relaxations possible, such as an extension of the total driving time to 10 hours or an extension of the duty time to 15 hours. However, these relaxations are only allowed for a limited number of times (e.g., the extension to 10 hours of driving time is only allowed 2 times a week). We show in Section 6 how to extend our ILP model to also handle these relaxations.

3. ILP formulation for the VDO

Since breaks can be taken both before and after serving a customer, we have to decide for every customer i at what time service starts and at what time the vehicle leaves the customer. Therefore, we introduce the variables X_i^s and X_i^d to indicate the start time of service at customer i and the departure time from customer i , respectively. In addition, we introduce the variables W_i^s and W_i^d to indicate the waiting time of the vehicle directly before and after serving customer i .

There are two types of breaks, namely breaks of at least b_{min}^1 hours and breaks of at least b_{min}^2 hours. Therefore, we introduce the variables $B_i^{p,l}$, indicating the break time at customer $i = 1, \dots, n$, before ($p = s$) or after ($p = d$) serving the customer, and of type $l = 1, 2$. To check whether a waiting time can be considered a break, we also introduce binary variables $Y_i^{p,l}$. If a realization of W_i^p does not exceed b_{min}^l , then the corresponding variables $Y_i^{p,l}$ and $B_i^{p,l}$ are set to zero. Otherwise, the corresponding variable $B_i^{p,l}$ takes the value of W_i^p .

Finally, to ensure that enough breaks are taken during and at the end of each driving period, we introduce binary variables V_{ij} ($j > i$). If a driving period starts at customer i and ends at customer j , then V_{ij} is set to 1. In that case, the break time at customer j must be at least b_{min}^1 , and the total break time at customers k ($i < k \leq j$) must be at least b_{total} . This results in the following ILP formulation:

$$\text{Min } X_{n+1}^s - X_0^d \tag{1}$$

$$X_i^s = X_{i-1}^d + c_{i-1}(X_{i-1}^d) + W_i^s \quad (i = 1, \dots, n + 1), \tag{2}$$

$$X_i^d = X_i^s + s_i + W_i^d \quad (i = 0, \dots, n), \tag{3}$$

$$X_i^s \geq e_i \quad (i = 0, \dots, n + 1), \tag{4}$$

$$X_i^s \leq l_i \quad (i = 0, \dots, n + 1), \tag{5}$$

$$W_i^p \geq b_{min}^l Y_i^{p,l} \quad (i = 1, \dots, n, l = 1, 2, p = s, d), \tag{6}$$

$$B_i^{p,l} \leq M Y_i^{p,l} \quad (i = 1, \dots, n, l = 1, 2, p = s, d), \tag{7}$$

$$B_i^{p,l} \leq W_i^p \quad (i = 1, \dots, n, l = 1, 2, p = s, d), \tag{8}$$

$$\sum_{k=0}^j c_k(X_k^d) \leq t_{dr} + M \sum_{k=1}^j V_{0k} \quad (j = 1, \dots, n), \tag{9}$$

$$\sum_{k=i}^j c_k(X_k^d) \leq t_{dr} + M \left(\sum_{k=i+1}^j V_{ik} + 1 - \sum_{k=0}^{i-1} V_{ki} \right), \tag{10}$$

$(i = 1, \dots, n - 1, j = i + 1, \dots, n),$

$$\sum_{j=1}^n V_{0j} \leq 1, \tag{11}$$

$$\sum_{j=i+1}^n V_{ij} \leq \sum_{k=0}^{i-1} V_{ki} \quad (i = 1, \dots, n - 1), \tag{12}$$

$$B_j^{s,1} + B_j^{d,1} \geq b_{min}^1 V_{ij} \quad (i = 0, \dots, n - 1, j = i + 1, \dots, n), \tag{13}$$

$$\sum_{k=i+1}^j (B_k^{s,2} + B_k^{d,2}) \geq b_{total} V_{ij} \quad (i = 0, \dots, n - 1, j = i + 1, \dots, n), \tag{14}$$

$$\sum_{k=0}^n c_k(X_k^d) \leq t_{max}, \tag{15}$$

$$\text{All variables} \geq 0, \tag{16}$$

$$Y_i^{p,l} \in \{0, 1\} \quad (i = 1, \dots, n, l = 1, 2, p = s, d), \tag{17}$$

$$V_{ij} \in \{0, 1\} \quad (i = 0, \dots, n - 1, j = i + 1, \dots, n). \tag{18}$$

The objective is to minimize a truck driver's duty time. Constraints (2) and (3) define the start time of service at and the departure time from each customer. Constraints (4) and (5) ensure that service starts in the given time window. Constraints (6) check whether a waiting period is enough to be considered a break. If not, then $Y_i^{p,l}$ is set to zero and Constraints (7) become tight. Constraints (8) ensure that the break time never exceeds the waiting time. Constraints (9) ensure that the first driving period does not exceed t_{dr} . If the total driving time between customers 0 and $j + 1$ exceeds t_{dr} ($\sum_{k=0}^j c_k(X_k^d) > t_{dr}$), then the first driving period must end at a customer $k, 0 < k < j + 1$ ($\sum_{k=1}^j V_{0k} = 1$). Constraints (10) ensure that the succeeding driving periods end in time. If a driving period starts at customer i ($\sum_{k=0}^{i-1} V_{ki} = 1$) and the total driving time between customers i and $j + 1$ exceeds t_{dr} ($\sum_{k=i}^j c_k(X_k^d) > t_{dr}$), then this driving period must end at a customer $k, i < k < j + 1$ ($\sum_{k=i+1}^j V_{ik} = 1$). Constraints (11) ensure that the first driving period ends at most once and Constraints (12) ensure that each succeeding driving period ends at most once. Constraints (13) ensure that a break of at least b_{min}^1 hours is taken at a customer at which a driving period ends and Constraints (14) ensure that in each driving period the total break time is at least b_{total} . Finally, Constraint (15) ensures that the total driving time does not exceed t_{max} . Note that the parameter M used in the model does not need to be very large, $M = l_{n+1} - e_0$ is sufficient.

So far, we have modeled the travel time function as a general function that depends on the time of departure. However, in general such a function cannot be written in proper ILP form. In Section 3.1, we model the time-dependent travel times as a continuous piecewise linear travel time function, and show how to write it in ILP form.

3.1. Travel time modeling

Several ways of modeling the time-dependent travel times have been proposed in the literature. Malandraki and Daskin (1992) propose a travel time step function. A disadvantage of this approach is that the non-passing property is not satisfied, i.e., if vehicles A and B traverse the same link in the network, and vehicle B departs later than vehicle A, but with a smaller travel time, then vehicle B could arrive earlier than vehicle A. Haghani and Jung (2005) propose a

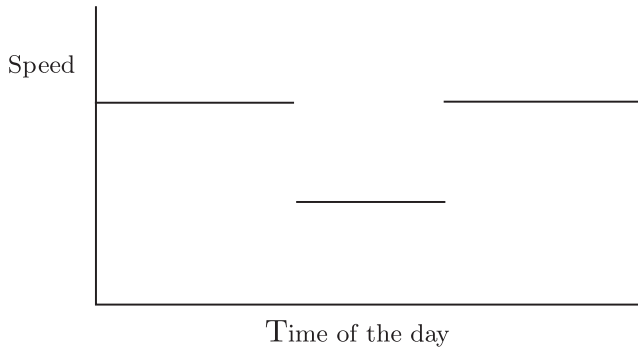


Fig. 1. Speed function.

continuous travel time function in which the slope is always greater than -1 . In that case, departing later can never result in an earlier arrival. The disadvantage of an arbitrary continuous travel time function is that it does not need to be (piecewise) linear. Therefore, we choose to follow the approach of Ichoua et al. (2003), who propose a travel speed step function for each link in the network. This approach results in a continuous piecewise linear travel time function. Since two vehicles traversing the same link drive with the same speed at any moment of time, the non-passing property is satisfied. Fig. 1 shows an example of a speed function; Fig. 2 presents the resulting travel time function.

Since the travel time function is piecewise linear, we can write it as m_i different functions $a_{i,r} + b_{i,r}(X_i^d - g_{i,r})$, where $g_{i,r}$, $r = 1, \dots, m_i$ indicate the times at which the slope of the travel time function changes. Furthermore, $a_{i,r}$ is the travel time at time $g_{i,r}$ and $b_{i,r}$ is the slope of the r th linear function. To determine in which interval $[g_{i,r}, g_{i,r+1}]$ the chosen departure time X_i^d falls, we introduce binary variables $U_{i,r}$ which take value one only if $g_{i,r} \leq X_i^d \leq g_{i,r+1}$. Next, we introduce variables $X_{i,r}^d$ which take the value of X_i^d if the corresponding variable $U_{i,r}$ is one, and zero otherwise. By replacing the function $c_i(X_i^d)$ by the variable C_i we derive the following ILP formulation to determine the travel time for departure time X_i^d :

$$\sum_{r=1}^{m_i} U_{i,r} = 1 \quad (i = 0, \dots, n), \quad (19)$$

$$g_{i,r} U_{i,r} \leq X_{i,r}^d \quad (i = 0, \dots, n, r = 1, \dots, m_i), \quad (20)$$

$$g_{i,r+1} U_{i,r} \geq X_{i,r}^d \quad (i = 0, \dots, n, r = 1, \dots, m_i), \quad (21)$$

$$\sum_{r=1}^{m_i} X_{i,r}^d = X_i^d \quad (i = 0, \dots, n), \quad (22)$$

$$C_i \geq a_{i,r} + b_{i,r}(X_i^d - g_{i,r}) + M(U_{i,r} - 1) \quad (i = 0, \dots, n, r = 1, \dots, m_i). \quad (23)$$

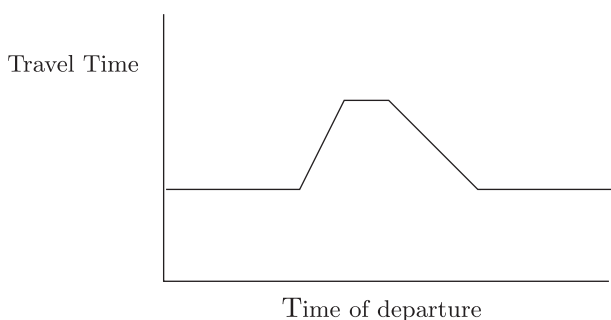


Fig. 2. Travel time function.

Constraints (19) ensure that exactly one $U_{i,r}$ takes value one. The $U_{i,r}$ with value one and Constraints (20) and (21) force the corresponding variable $X_{i,r}^d$ to be in the interval $[g_{i,r}, g_{i,r+1}]$, and all other variables $X_{i,r}^d$ to be zero. Constraints (22) force the only non-zero $X_{i,r}^d$ to equal X_i^d , and therefore $U_{i,r}$ can only take value one, if $g_{i,r} \leq X_i^d \leq g_{i,r+1}$. Finally, Constraints (23) are only tight if $U_{i,r}$ equals one, i.e., if $g_{i,r} \leq X_i^d \leq g_{i,r+1}$, which result in the required travel time functions.

4. Computational experiments

We set up the computational experiments as follows. First, we illustrate the potential duty time savings in practice by solving the VDO for a number of vehicle routes obtained from practice. Section 4.1 presents the results of this case study. Section 4.2 presents the results of testing the VDO on a set of routes obtained from best known solutions to the well-known Solomon (1987) instances for the VRPTW. These tests demonstrate the necessity of accounting for time-dependent travel times and driving hours regulations when constructing vehicle routes. Therefore, Section 5 proposes a best insertion heuristic for the VRPTW with time-dependent travel times and driving hours regulations. We implemented the solution methods and required data structures in Delphi 7, and solved the ILP using CPLEX 11 on a PC with a Core 2 Quad, 2.83 GHz CPU and 4 GB of RAM.

4.1. VDO: A case study

In order to test the practical impact of our solution approach for the VDO, we apply it to 12 vehicle routes provided by ORTEC. These vehicle routes are constructed for a Dutch client (of ORTEC) and contain between 12 and 36 customer visits per route (with an average of 21 visits). The routes are constructed by ORTEC's vehicle routing software SHORTREC, which contains various state of the art construction and improvement (local search) heuristics. These heuristics are adapted for practical use, implying that they account for several realistic constraints, such as time windows and driving hours regulations, and that the quality of solutions are measured in all relevant cost factors, such as number of vehicles used, total distance traveled, and total duty time. ORTEC has also implemented a greedy approach based on binary search to solve the VDO as a post-processing step of constructing the vehicle routes.

SHORTREC is often used in the Netherlands where traffic congestion regularly appears. SHORTREC's complete planning environment allows to account for traffic congestion to some extent, as may also be the case with other commercial vehicle routing software. However, traffic congestion is not accounted for during the vehicle route optimization phase, but in some post-processing phase in which planners can modify the routes by hand to improve them. Since information on time-dependent travel times could not be provided for the vehicle routes in this case study, we assume time-independent travel times. Driving hours regulations are accounted for during SHORTREC's vehicle route optimization phase. Therefore, feasible departure schedules exist for all vehicle routes in this case study.

Table 1 presents the duty times of the 12 routes before solving the VDO, after solving the VDO with ORTEC's greedy approach, and after solving it with our approach. The average reduction of the duty times by the greedy approach is 75 minutes. Our solutions reduce these duty times by an additional 32 minutes, on average. This implies that departure time optimization as a post-processing step of constructing the vehicle routes reduces duty times by 107 minutes, on average, which is 15% of the total duty time. In comparison with ORTEC's greedy approach, our approach reduces an additional 5.8% of the total duty time. Note that all other

Table 1
Duty times (minutes) for vehicle routes from practical case.

Route	VDO approach		ILP
	No	Greedy	
1	634	518	443
2	610	539	537
3	754	729	729
4	655	655	641
5	851	799	769
6	826	798	798
7	919	799	798
8	469	405	359
9	357	346	300
10	813	710	710
11	857	731	588
12	858	678	651

relevant cost factors (e.g., number of vehicles used, total distance traveled) remain the same. Therefore, these duty time reductions can be realized without introducing extra costs.

4.2. VDO: Solomon benchmarks

This section stresses the importance of incorporating time-dependent travel times and driving hours regulations in methods for constructing vehicle routes. For this purpose, we test our solution approach for the VDO on a selection of the 100-customer problem instances developed by Solomon (1987) for the VRPTW. We use those problem instances for which best known solutions identified by heuristics can be obtained from the literature. The routes obtained from these solutions form the problem instances for the VDO. Our preference was to test the VDO on routes obtained from good solutions to TDVRP instances, since these routes already account for time-dependent travel times. Unfortunately, the involved authors can no longer provide these routes (Ichoua et al., 2003; Fleischmann et al., 2004; Haghani and Jung, 2005). However, we shall demonstrate that even if one of the restrictions ‘time-dependent travel times’ or ‘driving hours regulations’ is neglected during the construction of the vehicle routes, then in many cases it is not possible to find feasible departure schedules. This implies that the routes are not applicable in practice.

The Solomon problem instances are categorized into 3 types of instances: *c*-instances in which customer locations are clustered, *r*-instances in which customers are randomly located in a square, and *rc*-instances in which 50% of the customers are clustered and 50% are randomly located. Each customer is given a hard time window in which its service must start. The time window at the depot indicates the earliest feasible departure time from the depot and the latest feasible return time at the depot. Furthermore, some of the problem instances have a relatively large time window at the depot and vehicles with a relatively large capacity, resulting in large vehicle routes (25–50 customers), while other instances have a relatively small time window at the depot, resulting in small vehicle routes (about 10 customers). Since the number of customers visited in a vehicle route defines the input size of the VDO, we discern small and large vehicle routes. This distinction allows us to investigate the impact of the input size of the VDO on the required computation time. The number of customers visited in a vehicle route ranges from 4 to 51 customers. We categorize the VDO problem instances into small (≤ 20 customers) and large (> 20 customers) problem instances.

The travel speed in the networks of the Solomon instances equals one. Therefore, the travel times in the Solomon instances equal the euclidean distances between the customer locations. Since the travel speed is time-independent, we develop speed

patterns, such that the average travel speed remains one. This methodology is similar to the one proposed by Ichoua et al. (2003). We define the time window at the depot from 6:30 am until 7:30 pm, which corresponds to a maximum daily working time of 13 hours. We assume that the morning traffic peak causes congestion from 7:00 am until 9:00 am, and the evening traffic peak from 5:00 pm until 7:00 pm. Furthermore, we discern light, medium, and heavy congestion. These three types of congestion cause speed drops during the peak hours of 25%, 50%, and 75%, respectively. Table 2 presents the resulting speed patterns. It turns out that with these speed patterns two of the selected Solomon instances (23 in total) contain some customer time windows that cannot be met, even not with a dedicated vehicle route. Therefore, we removed the routes obtained from best known solutions to these two Solomon instances from the problem set.

The VDO problem instances are composed of the vehicle routes resulting from best known solutions to the Solomon instances and the travel speed patterns in Table 2. Furthermore, we set $b_{min} = 0.25$, $b_{total} = 0.75$, $t_{dr} = 4.5$, and $t_{max} = 9$, corresponding to the European driving hours regulations. Since the original Solomon instances do not account for driving hours regulations nor time-dependent travel times, we investigate whether the developed routes allow feasible VDO solutions. Since we test the impact of two different realistic factors in vehicle routing, we develop two test scenarios: in Scenario 1 we do not consider driving hours regulations and in Scenario 2 we do consider driving hours regulations. In both scenarios, we solve the VDO for each of the three speed patterns as described before, as well as the case in which there are no speed drops at all. This allows us to also test the impact of driving hours regulations on vehicle routes in congestion free networks. Tables 3 and 4 present results on computation times and percentage of infeasible VRP routes by optimizing the departure times for Scenarios 1 and 2, respectively.

The computation times are small enough for practical use. The maximum computation time over all instances is 1.1 seconds (for Scenario 1 even 78 ms). Therefore, our approach to solve the VDO as a post-processing step of a VRPTW is feasible in practice. The number of variables in the ILP model and the LP bounds influence the computation times. The average and maximum number of variables for the VDO instances equal 549 and 2949, respectively. For the binary variables, these numbers are 301 and 2006, respectively. The gaps with the LP bounds vary from 1.37% on average when neither time-dependent travel times nor driving hours regulations are present, to 17.1% on average with heavy congestion and driving hours regulations.

The solution methods for the original VRP instances do not account for time-dependent travel times and driving hours regulations, and as a consequence the obtained routes are often too tight with respect to the time windows to schedule mandatory breaks. It generally holds that heavier traffic congestion results in fewer feasible vehicle routes. Therefore, vehicle routing methods should account for time-dependent travel times. However, this is not sufficient to obtain vehicle routes that can be used in practice under driving hours regulations. Tables 3 and 4 also show that about half of the routes that are feasible with respect to time-dependent travel times, but that ignore driving hours regulations, turn out to be infeasible when driving hour regulations are re-

Table 2
Speed patterns.

Type of congestion\time	6:30–7	7–9	9–17	17–19	19–19:30
Light	1.08	0.81	1.08	0.81	1.08
Medium	1.17	0.58	1.17	0.58	1.17
Heavy	1.27	0.32	1.27	0.32	1.27

Table 3
Results scenario 1: no driving hours regulations.

Problem size	# Instances	Congestion type	Average CPU (ms)	VRP route infeasible (%)
Small ^a	142	No	4	0.00
		Light	5	14.79
		Medium	4	45.07
		Heavy	2	69.01
Large ^b	25	No	18	0.00
		Light	26	16.00
		Medium	19	44.00
		Heavy	14	68.00
Average	167	No	6	0.00
		Light	8	14.97
		Medium	6	44.91
		Heavy	4	68.86

^a All routes in best known solutions to instances rc106 (Li and Lim, 2003), r107, r109, r111 and rc107 (Shaw, 1997), r108, r110 and rc105 (Berger and Barkaoui, 2004), and rc101, rc102, rc103, rc104 and rc108 (Czech and Czarnas, 2002).

^b All routes in best known solutions to instances r211 (Rochat and Taillard, 1995), and rc201, rc202, rc203, rc204, rc205, rc206, and rc207 (Czech and Czarnas, 2002).

Table 4
Results Scenario 2: with driving hours regulations.

Problem size	# Instances	Congestion type	CPU (s)	VRP route Infeasible (%)
Small	142	No	5	61.27
		Light	12	61.27
		Medium	13	76.06
		Heavy	9	85.21
Large	25	No	70	24.00
		Light	96	36.00
		Medium	156	56.00
		Heavy	146	68.00
Average	167	No	15	55.69
		Light	25	57.49
		Medium	34	73.05
		Heavy	29	82.63

spected. Therefore, these routes would fail in practice. This problem is clearly caused by the methods that build the vehicle routes; it does not affect the applicability of the VDO in practice. As we shall argue in the remainder of this section, it is not straightforward to overcome this problem.

First, slack time could be added to the original problem instances, such that time is reserved for scheduling mandatory breaks after the vehicle routes have been developed. To keep the proposed solution methods in the VRP literature directly applicable, this slack time should be spread out evenly over the travel times between (or service times at) the customers. We tested this approach by adding one sixth of slack travel time. At least one sixth of slack travel time is required, because the total travel time in a driving period does not exceed 4.5 hours, while 45 minutes of break time needs to be scheduled in such a period. Computational experiments show that this approach works well for light congestion (the percentage of infeasible vehicle routes reduces from 57.49% to 0.60%), but with medium and heavy congestion the percentage of infeasible routes remains rather large (11.98% and 49.10%, respectively). A drawback of this approach is that built-up slack might be lost when truck drivers have to wait at customers before they can start service. This is one of the reasons that many routes remain infeasible in case of medium and heavy congestion. Moreover, slack travel time may lead to suboptimal solutions.

Second, one could argue that the infeasibility problem is caused by the tightness of optimal solutions. We therefore also tested less

Table 5
Relation between tightness of VRPTW solution and VDO infeasibility.

Slack (hrs)		Best known solutions		Nearest neighbor solutions	
≥	<	# Routes	Infeasible (%)	# Routes	Infeasible (%)
0.00	0.02	53	98.11	15	100.00
0.02	0.04	65	75.38	30	100.00
0.04	0.06	34	50.00	21	61.90
0.06	0.08	11	46.36	20	44.00
0.08	0.10	1	0.00	9	11.11
0.10	0.12	2	0.00	6	16.67
0.12	0.14	0	–	13	23.08
0.14	–	1	0.00	53	0.00

sophisticated methods to develop the vehicle routes, resulting in worse VRP solutions with respect to the overall objective, but with possibly less tight routes with respect to the time windows. We tested this approach with a straightforward nearest neighbor heuristic. The results show that the percentage of infeasible vehicle routes decreases from 67.22% to 49.67% (in Scenario 2, averaged over all types of congestion), but the number of vehicle routes increases dramatically from 167 to 229. Although the number of feasible vehicle routes increases, the total number of customers in all feasible vehicle routes decreases from 739 to 661. It turns out that the nearest neighbor solutions contain some routes with many customers served, which are very tight and therefore are likely to result in infeasible VDO instances. Table 5 presents the relationship between the tightness of VRPTW solutions and the infeasibility of the resulting VDO instances (for medium traffic congestion). We measure tightness as the average difference between the earliest and latest feasible departure time over all nodes in a VRPTW route. This measure can be seen as the average slack in departure time from the nodes in a route. Table 5 shows that the infeasibility of the VDO clearly depends on the tightness of the VRPTW solution, independent of the used solution method. Therefore, the infeasibility of the VDO is not caused by a VRPTW solution method in particular, but results from the ignorance of time-dependent travel times and driving hours regulations when constructing the vehicle routes.

Therefore, since decomposition methods in which driving hours regulations and time-dependent travel times are only handled in a post-processing step fail, the need arises to develop new vehicle routing methods that account for time-dependent travel times and driving hours regulations. In the following section, we propose a vehicle routing method that integrates the ILP model of Section 3 with the construction of the vehicle routes.

5. An integrated solution approach

We propose an insertion heuristic for the VRPTW with time-dependent travel times and driving hours regulations. This construction heuristic is a first integrated approach for this problem. The insertion heuristic constructs a complete solution by sequentially inserting customers in the vehicle routes in the current partial solution, such that the increase in total duty time is minimal. For each vehicle route in the current partial solution and for each insertion position, we determine the feasibility and costs of inserting the customer by solving the ILP formulation of Section 3 for the new route. If the customer cannot be inserted in any of the vehicle routes in the current partial solution, a dedicated vehicle route is added to the partial solution.

Preliminary tests indicate that the order in which customers are inserted has a big impact on the solution quality. We tested ordering the customers by ascending time window opening time, descending time window opening time, ascending time window closing time, descending time window closing time, and order of

appearance in the input data. Inserting the customers in order of time window closing time leads to the best results in these tests. We therefore propose to use this ordering.

We test the insertion heuristic on the modified Solomon instances described in Section 4.2. If certain customer time windows cannot be met with one of the speed patterns, even not in a dedicated vehicle route, then we modify such time windows. If a time window closing time cannot be met, we increase it until it is not smaller than the earliest feasible arrival time for all speed patterns. We adjust time window opening times similarly. Appendix A provides a detailed description of the derivation of these test sets. We again use two test scenarios: Scenario 1 is *without* driving hours regulations (by relaxing the constraints considering these regulations), Scenario 2 is *with* driving hours regulations. This allows us to quantify the impact of driving hours regulations on vehicle routing.

Tables 6 and 7 present the results of Scenario 1 and 2, respectively. Driving hours regulations cause an increase in the number of vehicles used and the total duty time of about 5.7% and 4.3%, respectively. Accounting for traffic congestion, however, hardly has an impact on the number of vehicle routes and the total duty time for these instances, in which the average travel speed is the same for all speed patterns. Therefore, these test results show that it is possible to account for time-dependent travel times, resulting in feasible vehicle route plans, while maintaining the solution quality in terms of number of vehicles used and total duty time. Computation times, however, increase when traffic congestion is present.

In Scenario 1, computation times are small enough for practice (the maximum is 161 seconds over all instances). In Scenario 2, however, computation times for some problem instances explode to a maximum of 7.4 hours. It turns out that for problem instances with a few – but long – vehicle routes, computation times may become very large. Since 7.4 hours of computation time is not acceptable in practice, we propose to limit the computation times for

Table 6
Results insertion heuristic Scenario 1: no driving hours regulations.

Congestion type	Problem set	# veh.	Duty time	Average CPU (s)	Max CPU (s)
No	c1	11.0	11188	6	7
	c2	3.4	10632	34	50
	r1	15.5	3127	6	7
	r2	3.5	3009	40	71
	rc1	15.9	3358	5	6
	rc2	4.3	3501	27	59
	Average	9.1	5558	19	71
	Light	c1	10.9	11060	6
c2		3.5	10934	36	57
r1		15.5	3098	6	7
r2		3.5	3058	46	86
rc1		16.1	3473	6	6
rc2		4.1	3500	31	67
Average		9.1	5600	22	86
Medium		c1	10.7	11036	6
	c2	3.5	11082	36	58
	r1	15.9	3155	6	7
	r2	3.5	2972	57	161
	rc1	16.6	3624	5	6
	rc2	4.4	3515	33	69
	Average	9.3	5637	24	161
	Heavy	c1	10.7	11215	7
c2		4.0	12767	36	63
r1		15.3	3227	6	7
r2		3.5	3156	55	114
rc1		16.5	3559	5	6
rc2		4.4	3874	37	82
Average		9.3	5999	24	114

Table 7
Results insertion heuristic Scenario 2: with driving hours regulations.

Congestion type	Problem set	# veh.	Duty time	Average CPU (s)	Max CPU (s)
No	c1	11.0	11165	7	9
	c2	3.4	10693	50	116
	r1	16.8	3276	9	11
	r2	3.9	3390	126	467
	rc1	17.1	3620	8	12
	rc2	4.6	3648	623	4675
	Average	9.7	5728	125	4675
	Light	c1	10.9	11060	9
c2		3.5	10946	58	139
r1		16.2	3296	11	18
r2		3.8	3376	629	2268
rc1		16.4	3529	14	31
rc2		4.5	3681	1526	11722
Average		9.4	5741	356	11722
Medium		c1	10.7	11040	7
	c2	3.5	11106	59	162
	r1	16.3	3256	14	25
	r2	4.0	3346	2805	26643
	rc1	17.8	3712	9	14
	rc2	4.4	3685	1011	7621
	Average	9.7	5773	709	26643
	Heavy	c1	10.7	11211	8
c2		4.0	12761	65	204
r1		17.1	3434	34	113
r2		4.0	3512	1223	9601
rc1		19.0	3876	11	21
rc2		4.5	3949	343	2016
Average		10.1	6168	309	9601

solving each ILP. Next, we evaluate each insertion attempt with the best solution found after this maximum amount of computation time. Note that for each customer a dedicated vehicle route is feasible (the input is such that no travel-departure time combination leads to more than 4.5 hours of travel time). If the ILP solver does not find a feasible solution for a dedicated vehicle route within the allowed computation time, we set the duty time of this vehicle route to the depot opening hours (13 hours), which is a valid upper bound. This ensures that the method will always find a feasible solution (if we set the allowed ILP computation time to 0 we

Table 8
Results with different ILP solver time limits.

Time limit (s)	Congestion type	# veh.	Duty time	Average CPU (s)	Max. CPU (s)
∞	No	9.7	5728	125	4675
	Light	9.4	5741	356	11722
	Medium	9.7	5773	709	26643
	Heavy	10.1	6168	309	9601
1.0	No	9.7	5730	52	686
	Light	9.4	5744	99	1164
	Medium	9.7	5787	118	1705
0.5	Heavy	10.1	6190	123	1420
	No	9.7	5730	47	477
	Light	9.5	5747	76	721
0.25	Medium	9.7	5785	92	1008
	Heavy	10.1	6196	112	1231
	No	9.7	5741	43	332
0.1	Light	9.5	5784	59	398
	Medium	9.7	5800	73	619
	Heavy	10.2	6239	98	708
	No	11.2	6511	77	365
	Light	10.6	6830	70	272
	Medium	10.9	6526	87	387
	Heavy	13.3	7812	174	625

end up with n dedicated vehicle routes with maximum duty time each).

Table 8 presents the results for Scenario 2 for different ILP solver time limits, which limit the amount of computation time of each separate ILP. Average computation times decrease substantially with smaller time limits (from 6.2 minutes with unlimited ILP solver time to 1.6 minutes with a maximum of 1.0 second of ILP solver time per ILP); maximum computation times decrease dramatically (from 7.4 hours with unlimited ILP solver time to 28 minutes with a maximum of 1.0 second of ILP solver time per ILP). Decreasing the time limit even further mainly decreases the maximum computation times. The solution quality hardly decreases when the ILP solver time limit is set to 1.0 second (the number of vehicle routes and the total duty time increase by less than 0.5%). However, when the time limit is set too low (<0.25 seconds), the quality of the route plans substantially decreases (approximately 18% more vehicles and duty time when the ILP solver time limit is set to 0.1 second). Therefore, setting the time limit to 0.5 or 0.25 seconds gives a fair trade off between computation time and solution quality.

6. Model extensions

The ILP formulation proposed in Section 3 assumes one-day planning and that breaks are only taken at customers. There are several practical cases in which it is more convenient to extend the formulation to a multi-day planning or to assume that breaks can also be taken at parking lots. We demonstrate that these extensions can easily be incorporated in our ILP formulation.

For multi-day planning, some extra restrictions are imposed by the driving hours regulations. Both the European and North-American driving hours regulations impose a maximum on the total driving time and the total working time on a day, after which a rest has to be taken. More formally, after driving at most t_{max} hours and being on duty for at most d_{max} hours, a rest of at least t_{rest} hours has to be taken. Also, a maximum is imposed on the total driving and working time in an entire week. We show how the ILP formulation of Section 3 can be extended to one-week planning.

First, in Constraint (15), t_{max} must be replaced by the maximum driving time in a week. Next, to check whether a waiting time at a customer can be considered a rest, we introduce variables $B_i^{p,rest}$, $p = s, d$ and binary variables $Y_i^{p,rest}$, and we add the following constraints to the ILP formulation:

$$W_i^p \geq t_{rest} Y_i^{p,rest} \quad (i = 1, \dots, n, p = s, d), \quad (24)$$

$$B_i^{p,rest} \leq M Y_i^{p,rest} \quad (i = 1, \dots, n, p = s, d), \quad (25)$$

$$B_i^{p,rest} \leq W_i^p \quad (i = 1, \dots, n, p = s, d). \quad (26)$$

Next, we need to check whether the driving (duty) time does not exceed the maximum driving (duty) time on each day before a night's rest is taken. Therefore, we introduce the notion of daily period which has the following three properties: (1) Each daily period ends with a night's rest, (2) in each daily period the driving and duty time do not exceed the maximum driving and duty time, and (3) each time a daily period ends, a new daily period is initiated. Next, we introduce binary variables V_{ij}^{rest} which are set to 1 if a rest period starts at customer i and ends at customer j . To ensure that the driving time does not exceed the maximum driving time in each daily period, and each daily period ends with a rest of at least t_{rest} hours, we add the following constraints:

$$\sum_{k=0}^j c_k (X_k^d) \leq t_{max} + M \sum_{k=1}^j V_{0k}^{rest} \quad (j = 1, \dots, n), \quad (27)$$

$$\sum_{k=i}^j c_k (X_k^d) \leq t_{max} + M \left(\sum_{k=i+1}^j V_{ik}^{rest} + 1 - \sum_{k=0}^{i-1} V_{ki}^{rest} \right) \quad (i = 1, \dots, n-1, j = i+1, \dots, n), \quad (28)$$

$$\sum_{j=1}^n V_{0j}^{rest} \leq 1, \quad (29)$$

$$\sum_{j=i+1}^n V_{ij}^{rest} \leq \sum_{k=0}^{i-1} V_{ki}^{rest} \quad (i = 1, \dots, n-1), \quad (30)$$

$$B_j^{s,rest} + B_j^{d,rest} \geq t_{rest} V_{ij}^{rest} \quad (i = 0, \dots, n-1, j = i+1, \dots, n). \quad (31)$$

Ensuring that the duty time does not exceed the maximum duty time during each daily period can be done via similar constraints. The only difference is that waiting times and service times also add to the total duty time. Therefore, both the arrival time and the service completion time at each customer is a possible moment for exceeding the total duty time. Since there are two possible moments at each customer for starting (ending) a daily period, the total number of possible daily periods is four times the number of possible daily periods for the case with maximum driving time. Therefore, we need four times the number of binary variables V_{ij}^{rest} to indicate when a daily period starts and when it ends. Similarly, we need two times the constraints of type (27) and (30), and four times the constraints of type (28) and (31), to ensure that each daily period ends with a break of t_{rest} , the total duty time in the daily period does not exceed d_{max} , and each time a daily period ends, a new daily period is initiated.

To account for the possibility of extending the driving time twice a week, we add binary variables E_i , $i = 0, \dots, n$, which take value one if a new daily driving period starts at customer i , and the total driving time of this period can be extended to 10 hours. To ensure that the total number of daily driving time extensions does not exceed two, we add the constraint $\sum_{i=0}^n E_i \leq 2$. Next, we ensure that $E_i, i > 0$ can only take value 1 if a new daily driving period starts at customer i by adding constraints $E_i \leq \sum_{k=0}^{i-1} V_{ki}^{rest}$, $i = 1, \dots, n$. Finally, to allow for the driving time extensions of 1 hour, we adjust Constraints (27) and (28):

$$\sum_{k=0}^j c_k (X_k^d) \leq t_{max} + E_0 + M \sum_{k=1}^j V_{0k}^{rest} \quad (j = 1, \dots, n), \quad (32)$$

$$\sum_{k=i}^j c_k (X_k^d) \leq t_{max} + E_i + M \left(\sum_{k=i+1}^j V_{ik}^{rest} + 1 - \sum_{k=0}^{i-1} V_{ki}^{rest} \right) \quad (i = 1, \dots, n-1, j = i+1, \dots, n). \quad (33)$$

To incorporate the possibility of taking a break at parking lots along the route, we can simply model these parking lots as customers with zero service time and maximum time window (i.e., $[e_0, I_{n+1}]$).

7. Conclusions

We introduced the VDO and first approached it as a post-processing step of solving a VRPTW. We proposed an ILP formulation for the VDO which is flexible with respect to several practical extensions. This flexibility was demonstrated while writing this paper, as the European driving hours regulations changed and we were able to quickly adapt the ILP formulation to the new regulations.

The computational experiments show that the VDO can be solved to optimality within practical computation times. Furthermore, a case study demonstrates that optimizing departure times may lead to duty time reductions of 15%, on average. Also a greedy approach for the VDO that is used in practice could be improved by

6% duty time reductions, on average. Such duty time reductions imply significant cost savings for logistic service providers and distribution firms.

Finally, the computational experiments show that VRP routes will only be of practical use if driving hours regulations and time-dependent travel times are accounted for during the development of vehicle routes. We argued that this problem is only solved by developing new vehicle routing methods. Therefore, we proposed a first integrated approach for the VRPTW with time-dependent travel times and driving hours regulations using the ILP formulation for the VDO. The average computation times with this approach are small enough for practical use. However, for some problem instances that allow long vehicle routes (in terms of number of customers), computation times become very large. We resolved this issue by limiting the ILP solver time, resulting in substantial computation time reductions and allowing practical computation times for all problem instances, whilst maintaining the solution quality.

Acknowledgment

This work was financially supported by Stichting Transumo through the project ketensynchronisatie. We thank ORTEC for providing the case data. We also thank the anonymous referees for their helpful comments to improve this paper.

Appendix A. Test sets

We propose a test set for the VRPTW with time-dependent travel times and driving hours regulations derived from the original Solomon instances for the VRPTW. To include time-dependent travel times, we introduce speed patterns reflecting different levels of traffic congestion. We consider five different periods throughout the day reflecting the morning and evening peak periods and the three periods before, between, and after these peak periods. During each period, the travel speed is constant and the same for each arc in the customer network. We assume the morning peak to last from 7:00 am until 9:00 am, and the evening traffic peak from 5:00 pm until 7:00 pm. Additionally, we define the time window at the depot from 6:30 am until 7:30 pm, corresponding to a maximum daily working time of 13 hours. For this purpose, we scale the original depot opening hours. We scale all other time windows, as well as the driving distances, accordingly. We discern light, medium, and heavy congestion, causing speed drops during the peak hours of 25%, 50%, and 75%, respectively. We normalize the speed patterns, such that the average speed over the day is one for each speed pattern. Table 2 presents the resulting speed patterns.

Due to the speed drops during the peak hours, it may happen that a certain customer time window cannot be met under one of the speed patterns, even not in a dedicated vehicle route. In such cases, we modify these time windows as follows. If a time window closing time cannot be met due to a late arrival under one of the speed patterns, then we set this time window closing time equal to the maximum earliest arrival time over all speed patterns. Similarly, if a time window opening time is so late that the vehicle cannot return in time to the depot under one of the speed patterns, then we modify this opening time such that it equals the minimum latest feasible start service time at this customer over all speed patterns.

References

- Archetti, C., Savelsbergh, M.W.P., 2009. The trip scheduling problem. *Transportation Science* 43 (1), 417–431.
- Baràn, B., Schaerer, M., 2003. A multiobjective ant colony system for vehicle routing problem with time windows. In: *Proceedings of the 21st IASTED International Conference on Applied Informatics*, pp. 97–102.
- Berger, J., Barkaoui, M., 2004. A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research* 31 (12), 2037–2053.
- Czech, Z.J., Czarnas, P., 2002. Parallel simulated annealing for the vehicle routing problem with time windows. In: *10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pp. 376–383.
- Eglese, R.W., Maden, W., Slater, A., 2006. A road timetable™ to aid vehicle routing and scheduling. *Computers & Operations Research* 33 (12), 3508–3519.
- European Union, 2006. Regulation (EC) No 561/2006 of the European parliament and of the council of 15 March 2006 on the harmonisation of certain social legislation relating to road transport and amending council regulations (EEC) No 3821/85 and (EC) No 2135/98 and repealing council regulation (EEC) No 3820/85. *Official Journal of the European Union* L 102/1, 11–04–2006.
- Federal Motor Carrier Safety Administration, 2008. Hours-Of-Service Regulations, <<http://www.fmcsa.dot.gov/rules-regulations/topics/hos/index.htm>>.
- Fleischmann, B., Gietz, M., Gnutzmann, S., 2004. Time-varying travel times in vehicle routing. *Transportation Science* 38 (2), 160–173.
- Geiger, M.J., 2001. Genetic algorithms for multiple objective vehicle routing. In: *Metaheuristics International Conference 2001 (MIC'2001)*, pp. 348–353.
- Goel, A., 2009. Vehicle scheduling and routing with drivers' working hours. *Transportation Science* 43 (1), 17–26.
- Goel, A., Kok, A.L., 2009a. Efficient scheduling of team truck drivers in the European Union, Working Paper, University of Leipzig.
- Goel, A., Kok, A.L., 2009b. Efficient truck driver scheduling in the United States, Working Paper, University of Leipzig.
- Gromicho, J., 2008. Personal communication, Head of Algorithmic Development ORTEC.
- Haghani, A., Jung, S., 2005. A dynamic vehicle routing problem with time-dependent travel times. *Computers & Operations Research* 32 (11), 2959–2986.
- Hill, A.V., Benton, W.C., 1992. Modelling intra-city time-dependent travel speeds for vehicle scheduling problems. *The Journal of the Operational Research Society* 43 (4), 343–351.
- Hong, S.C., Park, Y.B., 1999. A heuristic for bi-objective vehicle routing with time window constraints. *International Journal of Production Economics* 62 (3), 249–258.
- Ichoua, S., Gendreau, M., Potvin, J.Y., 2003. Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research* 144 (2), 379–396.
- Jozefowicz, N., Semet, F., Talbi, E.G., 2008. Multi-objective vehicle routing problems. *European Journal of Operational Research* 189 (2), 293–309.
- 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*, in press, doi:10.1287/trsc.1100.0331.
- Li, H., Lim, A., 2003. Local search with annealing-like restarts to solve the VRPTW. *European Journal of Operational Research* 150 (1), 115–127.
- Malandraki, C., Daskin, M.S., 1992. Time dependent vehicle routing problems: Formulations properties and heuristic algorithms. *Transportation Science* 26 (3), 185–200.
- Rochat, Y., Taillard, É.D., 1995. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics* 1 (1), 147–167.
- Savelsbergh, M.W.P., 1992. The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing* 4 (2), 146–154.
- Sessomboon, W., Watanabe, K., Irohara, T.Y.K., 1998. A study on multi-objective vehicle routing problem considering customer satisfaction with due-time (the creation of pareto optimal solutions by hybrid genetic algorithm). *Transaction of the Japan Society of Mechanical Engineering*.
- Shaw, P., 1997. A new local search algorithm providing high quality solutions to vehicle routing problems, Working paper, University of Strathclyde, Glasgow, Scotland.
- Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* 35 (2), 254–265.
- Toth, P., Vigo, D., 2002. *The vehicle routing problem*, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia.
- Van Woensel, T., Kerbache, L., Peremans, H., Vandaele, N., 2008. Vehicle routing with dynamic travel times: A queueing approach. *European Journal of Operational Research* 186 (3), 990–1007.
- Xu, H., Chen, Z.L., Rajagopal, S., Arunapuram, S., 2003. Solving a practical pickup and delivery problem. *Transportation Science* 37 (3), 347–364.