CIRRELT

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

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

The Time-Dependent Shortest Path and Vehicle Routing Problem

Marjolein Veenstra
Leandro C. Coelho

August 2017

CIRRELT-2017-57

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2017-012.

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

The Time-Dependent Shortest Path and Vehicle Routing Problem

Marjolein Veenstra[1], Leandro C. Coelho[1,2,*]

[1] University of Groningen, Faculty of Economics and Business, Department of Operations, Nettelbosje 2, 9747 AE Groningen, The Netherlands

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Operations and Decision Systems, 2325 de la Terrasse, Université Laval, Québec, Canada G1V 0A6

Abstract. In this paper we introduce the time-dependent shortest path and vehicle routing problem. In this problem, a set of homogeneous vehicles is used to visit a set of customer locations dispersed over a very large network, such as the travel times between any two customers must be computed as a time-dependent shortest path problem. The travel time of each arc is time-dependent and therefore the shortest path between two locations changes over time. The aim of the problem is to simultaneously determine the sequence in which the customer locations are visited and the arcs traveled on the paths between each pair of consecutively visited customers, such that the sum of the arrival times of the vehicles back at the depot is minimized. We are the first to formally define and solve this fully integrated problem, giving bounds to it. We show that one must consider the whole underlying street network in order to truly obtain the fastest paths between two points. We test our formulation on a set of real-life instances generated from a dataset of the road network in Québec City, Canada. Our results indicate that neglecting traffic can impose substantial delays for the visits, which would require more trucks and more mileage to perform the same deliveries. Our work adds a new research avenue to city logistics and congestion/emission studies.

Keywords. Time-dependent vehicle routing problem, time-dependent shortest path problem, green logistics, city logistics.

Acknowledgements. This project was funded by the Dutch Institute for Advanced Logistics (Dinalog) and partly funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 2014-05764. This support is greatly acknowledged. We thank our industrial partner for providing data.

---

* Corresponding author: Leandro.Coelho@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2017

1 Introduction

Distribution companies performing several deliveries per day across the city have to deal with congestion and delays when planning their routes. Therefore, they need to incorporate time-dependent travel times in their routing tools. Time-dependent travel times imply that the time it takes to traverse an arc depends on the departure time at the starting node of the arc. As a result, the shortest path between two delivery locations can change depending on the departure time at the origin. Consider the road network as presented in Figure 1, consisting of six different road intersections: A, B, C, D, E, and F. Suppose a vehicle consecutively visits delivery locations A and F. Then, there exist three paths from location A to location F, namely (A, B, D, F), (A, B, E, F) and (A, C, E, F). The travel time of the arcs between the intersections is time-dependent and therefore the shortest path between locations A and F can change between these three paths throughout the day. Hence, planning the vehicle routes includes not only assigning customers to vehicles and deciding upon the sequence in which the customer locations are visited, but also determining the arcs which are traveled between two consecutive delivery locations. Obviously, in large networks there are much more than three ways to travel between two customers, and also the number of customers to visit and their sequence can be very large.

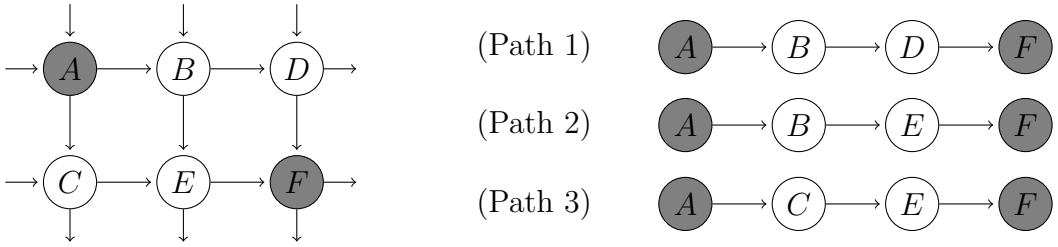


Figure 1: On the left, the graphical representation of a part of a road network with one-way streets and six intersections. On the right, the three possible paths from location A to location F.

In this paper, we introduce the time-dependent shortest path and vehicle routing problem

(TDSPVRP). In the TDSPVRP, a homogeneous fleet of capacitated vehicles based at the depot is used to visit a set of customer locations. In this problem, we simultaneously determine the sequence in which the customer locations are visited by the vehicles and the arcs traveled on the paths between two consecutively visited customers. The travel time of each arc is time-dependent and therefore the shortest path between two locations, with respect to travel time, is dependent on the departure time at the first location. The goal of the problem is to find feasible routes that specify the sequence in which the customers are visited and the paths between those customers, such that each customer is *served* exactly once. Note that a customer node can be visited more than once because its node is on the path leading to another customer. The objective is to minimize the sum of the arrival times of the vehicles back at the depot. Hence, instead of minimizing driving costs, we focus on avoiding congestion and traffic resulting in running at more reasonable speeds, which in turn results in less fuel consumption and less emissions. This objective is in line with recent research on routing in a city logistics context. For example, Koç et al. [13] that consider depot location, fleet composition and routing in a city logistics context and define the routing cost in terms of fuel consumption and CO_2 emissions, and Bektaş and Laporte [1], Franceschetti et al. [8] and Koç et al. [12] that each consider a specific pollution-routing problem. The pollution-routing problem is a variant of the vehicle routing problem with an objective that not only considers travel distance, but also accounts for the amount of greenhouse gas emissions, fuel, travel times and their costs. Also from a practical point of view this objective is of interest, since it aims at avoiding congestion for all vehicles. By considering a finite time horizon, we prevent that too much of the workload is assigned to only a small set of the vehicles.

The time-dependent vehicle routing problem (TDVRP) is a vehicle routing problem with time-dependent travel times but with only one path linking any two customers. Hence, solving the TDVRP does not require the time-dependent shortest path aspect to be present. Therefore, the travel times between two nodes, i.e., customer or depot nodes, are supposedely known and are input to the problem. As a result, the graphs for the TDVRP

are very dense, i.e., a complete graph where all nodes need to be visited and the only arcs that are present are the arcs that connect each pair of customer nodes and each customer node with the depot node. This is in contrast with a TDSPVRP instance that requires the complete road network as input. As a consequence, TDSPVRP graphs are relatively sparse, with many nodes and relatively few of them, i.e., the customer nodes, that need to be visited. Figure 2 illustrates two TDSPVRP instances, containing 10 and 50 customers. Note that the TDSPVRP can be converted into a TDVRP by solving *all pairs* of shortest paths for *each* starting time. This procedure is, however, very time consuming and prohibitive. For this reason, we formally and explicitly define the TDSPVRP.

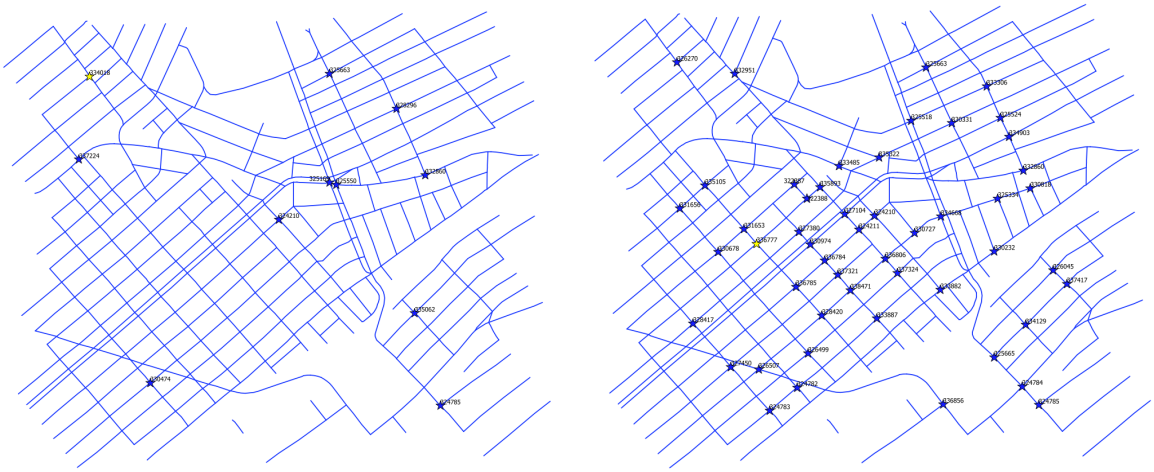


Figure 2: Instances with 10 customers (left) and 50 customers (right); the depot and customers locations are marked with stars.

The TDSPVRP is a combination of the time-dependent shortest path problem (TDSPP) and the TDVRP. The TDSPP is a shortest path problem with time-dependent travel times and was introduced by Cooke and Halsey [4]. Their algorithm was quickly improved by Dreyfus [7]. Other algorithms for the TDSPP are developed by, e.g., Chabini [2], Ding et al. [5], Orda and Rom [16] and Ziliaskopoulos and Mahmassani [18]. The TDVRP was introduced by Malandraki and Daskin [15] who proposed a mixed integer formulation for the problem. Many have formulated the problem exactly, e.g., Chen et al. [3] and Soler et al. [17] who have proposed a conversion of the TDVRP into a larger graph defining

a simple capacitated vehicle routing problem. Several heuristics are proposed for the problem, e.g., Donati et al. [6] who proposed a multi-ant colony system and Hashimoto et al. [9] who developed an iterated local search heuristic. Related to our research are the papers of Kok et al. [14] and Huang et al. [10]. Kok et al. [14] consider the combination of time-dependent shortest path problems and time-dependent vehicle routing problems. They use a restricted dynamic programming heuristic to solve four different combinations of problems and show significant improvements when considering time-dependent shortest paths in the TDVRP. Huang et al. [10] consider the time-dependent vehicle routing problem with path flexibility (TDVRP-PF). In the TDVRP-PF, path selection is explicitly considered and integrated in the TDVRP. This means that each arc between two customer nodes has multiple corresponding paths in the underlying road network [10]. The path selection decision is based on the departure time at the customer node and the level of congestion. For each pair of customers a small set of candidate paths are preprocessed, based on the knowledge of the network. The authors formulate the deterministic version of the TDVRP-PF as a mixed integer program and the stochastic version as a two-stage stochastic mixed integer program. Results show that path flexibility provides significant savings. In the formulations of Huang et al. [10] only the candidate paths can be selected. The advantage of our approach is that all paths between two customers can be used and therefore a better solution might be found. Moreover, our approach is applicable to general networks, for which no usage information is known or available. Finally, embedding the decision on the arcs traveled between two consecutive delivery locations in the mixed integer program ensures that the shortest paths do not have to be preprocessed. Solving the shortest paths between all pairs of customers for each starting time is very time consuming.

In this paper we develop an exact method, with the aim of providing and improving dual bounds for the problem. Given the size and difficulty of the problem, heuristics should provide much tighter primal bounds than our exact method. The goal of this paper is to provide the first set of benchmark instances for the TDSPVRP and to give bounds to the

problem, opening a research avenue for others. The remainder of this paper is as follows. In Section 2 we give the mathematical formulation for the TDSPVRP and we develop inequalities to strengthen this formulation and to improve the lower bounds. Section 3 reports the results from the experiments performed on real-life instances from a dataset of the road network in Québec City, discusses the impact of the valid inequalities and provides insights on time-dependent optimization. Conclusions are given in Section 4.

2 Problem statement and mathematical formulation

The TDSPVRP is defined on a directed time-dependent graph $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{H})$, where $\mathcal{N}$ is the set of nodes, $\mathcal{A}$ is the set of arcs and $\mathcal{H}$ is the set of time periods. Let $\mathcal{N} = \mathcal{N}' \cup \mathcal{N}_p$, where we refer to $\mathcal{N}'$ as the set of location nodes and $\mathcal{N}_p$ as the set of intermediate nodes. Let the set of location nodes $\mathcal{N}' = \mathcal{N}_c \cup \mathcal{N}_o \cup \mathcal{N}_d$, where $\mathcal{N}_c$ is the set of customer nodes, $\mathcal{N}_o$ is the set of origin depot nodes and $\mathcal{N}_d$ is the set of destination depot nodes. The origin and destination depot nodes are duplicated such that there is a unique origin and destination depot node for each vehicle from the homogeneous set $\mathcal{K} = \{1, \ldots, K\}$. For modeling purposes, we created for each vehicle an arc between its origin and destination depot node. We set the travel times of these K arcs equal to zero for each time period. Traveling along this arc represents not using this vehicle, but satisfying other constraints such as the ones imposing that every vehicle must leave the origin depot and arrive at the destination depot. Let $\mathcal{N}_p$ be the set of nodes that can be visited on a path between two location nodes, including nodes $i \in \mathcal{N}'$. The location nodes $\mathcal{N}'$ need to be visited exactly once, whereas the intermediate nodes do not necessarily have to be visited, but can be visited once or multiple times. Note that visiting an intermediate node $j \in \mathcal{N}_p$ that is a copy of a location node $i \in \mathcal{N}'$, corresponds to driving by this location without serving it. Let $\mathcal{H} = \{1, \ldots, H\}$ define H time periods, with each time period representing a time slot of length $\overline{T}$. We assume that the travel time of arc $(i, j) \in \mathcal{A}$ starting in time period h is known and is equal to t_{ijh}. The service time of a node $i \in \mathcal{N}'$ is given by s_i, where

$s_i = 0\ \forall i \in \mathcal{N}_o \cup \mathcal{N}_d$ and $s_i > 0\ \forall i \in \mathcal{N}_c$. Waiting time is allowed at each node, which might be beneficial if the FIFO property does not hold, i.e., if it may occur that when traveling arc (i, j), postponing the departure time at node i results in an earlier arrival time at node j. The demand at a node is given by $d_i\ \forall i \in \mathcal{N}'$, where $d_i = 0\ \forall i \in \mathcal{N}_o \cup \mathcal{N}_d$ and $d_i > 0\ \forall i \in \mathcal{N}_c$. The capacity of each vehicle $k \in \mathcal{K}$ is equal to Q. The goal of the problem is to determine vehicle routes satisfying the demands of all customers, creating proper paths between two consecutive customer visits, such that the total of the arrival time of the vehicles back to the depot is minimized.

In the definition of our variables, we use the term *shortest path towards a node*. If $i \in \mathcal{N}'$ is the last location node visited before the visit to location node $j \in \mathcal{N}'$, then the nodes on the shortest path towards node j are nodes i, j, and all intermediate nodes visited between nodes i and j, and the arcs on the shortest path towards node j are all arcs that are traversed between nodes i and j. An example is given in Figure 3, which depicts a partial route $\mathcal{R} = (1, 2, 3, 4, 5, 6)$, where the location nodes are given by $\mathcal{N}' = \{1, 3, 6\}$ and the intermediate nodes are in the set $\mathcal{N}_p = \{2, 4, 5\}$. The nodes and arcs on the shortest path towards node 3 are given by $\{1, 2, 3\}$ and $\{(1, 2),\ (2, 3)\}$, respectively. The nodes and arcs on the shortest path towards node 6 are given by $\{3, 4, 5, 6\}$ and $\{(3, 4),\ (4, 5),\ (5, 6)\}$, respectively.

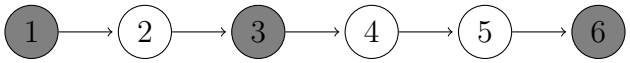


Figure 3: Partial route with $\mathcal{N}' = \{1, 3, 6\}$ and $\mathcal{N}_p = \{2, 4, 5\}$.

Our mathematical model is defined as follows. Let x_{ij} be a binary variable equal to one if and only if node $i \in \mathcal{N}'$ is the last location node visited before location node $j \in \mathcal{N}'$ and y_{ijl} be a binary variable equal to one if and only if arc $(i, j) \in \mathcal{A}$ is traversed on the shortest path towards node $l \in \mathcal{N}'$. Hence, in Figure 3, we have $x_{13} = x_{36} = 1$, $x_{ij} = 0$ otherwise, and $y_{123} = y_{233} = y_{346} = y_{456} = y_{566} = 1$, $y_{ijl} = 0$ otherwise. Let $T_{il} \in \mathbb{Z}$ be the time at which the vehicle departs from node $i \in \mathcal{N}$ on the shortest path towards node $l \in \mathcal{N}'$, and $q_i \in \mathbb{Z}$ reflect the cumulated load carried by the vehicle departing from node

$i \in \mathcal{N}'$. The binary variable δ_{plh} is equal to one if and only if a vehicle leaves node p on the shortest path towards node $l \in \mathcal{N}'$ in time period $h \in \mathcal{H}$. Let $\delta^2_{pml} \in \mathbb{Z},\ p \in \mathcal{N},\ m \in \mathcal{N}_p \cup l : (p,m) \in \mathcal{A},\ l \in \mathcal{N}'$ be the time at which a vehicle leaves node p on the shortest path towards node l if arc (p,m) is traversed on the shortest path towards node l, zero otherwise. Let $\delta^3_{pmlh} \in \mathbb{Z},\ p \in \mathcal{N},\ m \in \mathcal{N}_p \cup l : (p,m) \in \mathcal{A},\ l \in \mathcal{N}',\ h \in \mathcal{H}$ be the travel time of arc (p,m) in time period h if arc (p,m) is traversed on the shortest path towards node l in time period h, zero otherwise.

The model can then be defined as follows:

$$\min \quad \sum_{i \in \mathcal{N}_d} T_{ii} \tag{1}$$

s.t.
$$\sum_{i \in \mathcal{N}'} x_{ij} = 1 \qquad \forall j \in \mathcal{N}' \backslash \mathcal{N}_o \tag{2}$$

$$\sum_{j \in \mathcal{N}'} x_{ij} = 1 \qquad \forall i \in \mathcal{N}' \backslash \mathcal{N}_d \tag{3}$$

$$\sum_{j:(i,j) \in \mathcal{A}} y_{ijl} - \sum_{j:(j,i) \in \mathcal{A}} y_{jil} = \begin{cases} 0 & \text{if } i \notin \mathcal{N}' \\ -1 & \text{if } i = l \\ x_{il} & \text{if } i \in \mathcal{N}' \backslash l \end{cases} \qquad \forall i \in \mathcal{N},\ l \in \mathcal{N}' \backslash \mathcal{N}_o \tag{4}$$

$$x_{ij} - \sum_{p:(i,p) \in \mathcal{A}} y_{ipj} = 0 \qquad \forall i,\ j \in \mathcal{N}' \tag{5}$$

$$q_j = 0 \qquad \forall j \in \mathcal{N}_o \tag{6}$$

$$q_i - q_j + Q x_{ij} + (Q - d_i - d_j)\, x_{ji} \leq Q - d_i \qquad \forall i,\ j \in \mathcal{N}_c : i \neq j \tag{7}$$

$$q_j \leq Q \qquad \forall j \in \mathcal{N}_d \tag{8}$$

$$T_{ml} \geq T_{mm} - H\overline{T}\,(1 - x_{ml}) \qquad \forall l \in \mathcal{N}',\ m \in \mathcal{N}' \backslash \{\mathcal{N}_o \cup l\} \tag{9}$$

$$T_{ml} \geq \sum_{p:(p,m) \in \mathcal{A}} \left(\delta^2_{pml} + \sum_{h \in \mathcal{H}} \delta^3_{pmlh} \right) \qquad \forall l \in \mathcal{N}',\ m \in \mathcal{N}_p \tag{10}$$

$$T_{ml} \geq \sum_{p:(p,m) \in \mathcal{A}} \left(\delta^2_{pml} + \sum_{h \in \mathcal{H}} \delta^3_{pmlh} \right) + s_l \qquad \forall l \in \mathcal{N}',\ m = l \tag{11}$$

$$\sum_{h\in\mathcal{H}} \delta_{llh} = 1 \qquad \forall l \in \mathcal{N}' \tag{12}$$

$$\sum_{h\in\mathcal{H}} \delta_{ilh} = \sum_{j:(i,j)\in\mathcal{A}} y_{ijl} \qquad \forall i \in \mathcal{N},\ l \in \mathcal{N}'\backslash i \tag{13}$$

$$T_{pl} \geq (h-1)\,\overline{T}\delta_{plh} \qquad \forall p \in \mathcal{N},\ l \in \mathcal{N}',\ h \in \mathcal{H} \tag{14}$$

$$T_{pl} + 1 \leq H\overline{T}\,(1-\delta_{plh}) + h\overline{T} \qquad \forall p \in \mathcal{N},\ l \in \mathcal{N}',\ h \in \mathcal{H} \tag{15}$$

$$\delta^2_{pml} \geq T_{pl} - (1-y_{pml})\,H\overline{T} \qquad \forall p \in \mathcal{N},\ m \in \mathcal{N}_p \cup l : (p,m) \in \mathcal{A}, l \in \mathcal{N}' \tag{16}$$

$$\delta^3_{pmlh} \geq t_{pmh}\,(y_{pml} + \delta_{plh}) - t_{pmh} \qquad \forall p \in \mathcal{N},\ m \in \mathcal{N}_p \cup l : (p,m) \in \mathcal{A}, l \in \mathcal{N}',\ h \in \mathcal{H} \tag{17}$$

$$x_{ij} \in \{0,1\} \qquad \forall i,\ j \in \mathcal{N}' \tag{18}$$

$$y_{ijl} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A},\ l \in \mathcal{N}' \tag{19}$$

$$T_{il} \in \mathbb{Z} \qquad \forall i \in \mathcal{N},\ l \in \mathcal{N}' \tag{20}$$

$$q_i \in \mathbb{Z} \qquad \forall i \in \mathcal{N}' \tag{21}$$

$$\delta_{plh} \in \{0,1\} \qquad \forall p \in \mathcal{N},\ l \in \mathcal{N}', h \in \mathcal{H} \tag{22}$$

$$\delta^2_{pml} \in \mathbb{Z} \qquad \forall p \in \mathcal{N},\ m \in \mathcal{N}_p \cup l : (p,m) \in \mathcal{A}, l \in \mathcal{N}' \tag{23}$$

$$\delta^3_{pmlh} \in \mathbb{Z} \qquad \forall p \in \mathcal{N},\ m \in \mathcal{N}_p \cup l : (p,m) \in \mathcal{A}, l \in \mathcal{N}',\ h \in \mathcal{H}. \tag{24}$$

The objective (1) is to minimize the sum of the arrival times of the vehicles at the depot. Constraints (2) and (3) are standard degree constraints. Constraints (4) and (5) ensure that $y_{ijl} = 1$ if arc (i,j) is on the shortest path towards node l, and 0 otherwise. Constraints (6)–(8) are the load constraints. The explanation is based on customers that require items to be picked, but the constraints can also be applied when dealing with customers that need items to be delivered. The formulation can be easily adapted to

account for both pickup and delivery customers. Constraints (6) ensure that the vehicles leave the depot empty, constraints (7) update the load variables at the location nodes, and constraints (8) ensure that the vehicle capacity is never exceeded. Constraints (9)–(11) update the timing variable T_{ml}, where there are three possibilities: (1) if m is the first node on the shortest path towards node l, then due to constraints (9) T_{ml} is at least the departure time at node m on the shortest path towards node m, (2) if m is an intermediate node on the shortest path towards node l, then due to constraints (10) T_{ml} is at least the departure time at the node visited before node m on the shortest path towards node l added to the time-dependent travel time between these nodes, and (3) if m is the last node on the shortest path towards node l, i.e., $m = l$, then due to constraints (11) T_{ml} is at least the departure time at the node visited before node m on the shortest path towards node l added to the time-dependent travel time between these nodes added to the service time of node l. Note that the greater than or equal to signs in constraints (9)–(11) ensure that waiting time is allowed. Constraints (12)–(15) link paths and times via variable δ. More specifically, constraints (12) ensure that there is only one time interval associated with the departure time at a location node. Constraints (13) ensure that there is only one time period associated with the departure time at a node $i \neq l$ on the shortest path towards node l, and $\delta_{ilh} = 0\ \forall h \in \mathcal{H}$ if node i is not on the shortest path towards node l. Constraints (14)–(15) ensure that the departure time at node i on the shortest path towards node l is within the bounds of the time interval h for which $\delta_{ilh} = 1$. Constraints (16) define the variable δ^2_{pml}, which equals the time at which a vehicle leaves node p on the shortest path towards node l if arc (p, m) is used on the shortest path towards node l, and zero otherwise. Constraints (17) ensure that the variable δ^3_{pmlh} equals the travel time on arc (p, m) in time period h if the arc is traversed on the shortest path towards node l and a vehicle departs from node p on the shortest path towards node l in time period h, and zero otherwise. Constraints (18)–(24) set the range and nature of the variables.

The following valid inequalities can be imposed to strengthen the formulation.

$$y_{ijl} = 0 \qquad \forall (i,j) \in \mathcal{A},\ l \in \mathcal{N}_o \qquad (25)$$

$$H\overline{T} \sum_{j:(i,j)\in\mathcal{A}} y_{ijl} \geq T_{il} \qquad \forall i \in \mathcal{N},\ l \in \mathcal{N}'\backslash i \tag{26}$$

$$H\overline{T} y_{ijl} \geq \delta^2_{ijl} \qquad \forall (i,j) \in \mathcal{A},\ l \in \mathcal{N}' \tag{27}$$

$$\delta^3_{ijlh} \leq t_{ijh} \qquad \forall (i,j) \in \mathcal{A},\ l \in \mathcal{N}',\ h \in \mathcal{H} \tag{28}$$

$$\sum_{\substack{j\in\mathcal{N}_d \\ m\in\mathcal{N}\backslash\mathcal{N}_o:(m,j)\in\mathcal{A}}} y_{mjj} \geq \left\lceil \sum_{i\in\mathcal{N}'} d_i/Q \right\rceil \tag{29}$$

$$T_{ll} \geq T_{il} + s_l \qquad \forall l \in \mathcal{N}',\ i \in \mathcal{N}\backslash l \tag{30}$$

$$q_i \leq Q - \left(Q - \max_{j\in\mathcal{N}_c: j\neq i}\{d_j\} - d_i\right) \sum_{k\in\mathcal{N}_o} x_{ki} - \sum_{j\in\mathcal{N}_c} d_j x_{ij} \quad \forall i \in \mathcal{N}_c. \tag{31}$$

Constraints (25) ensure that there are no arcs on the shortest paths towards the origin nodes. Due to constraints (26), if node i is not on the shortest path towards location node l, then $T_{il} = 0$. Constraints (27) and (28) ensure that if arc $(i,j) \in \mathcal{A}$ is not used on the shortest path towards location node l, then $\delta^2_{ijl} = 0$ and $\sum_{h\in\mathcal{H}} \delta^3_{ijlh} = 0$, respectively. Constraint (29) states that the number of vehicles used is at least the total demand divided by the capacity of a vehicle. Constraints (30) ensure that the departure time at a location node l on the shortest path towards node l, is at least the departure time of each of the other nodes on the shortest path towards node l added to the service time of node l. Constraints (31) are valid inequalities, as proven in Kara et al. [11].

To better improve the bound and create other valid inequalities, let S_{ij} be the shortest path from node $i \in \mathcal{N}'$ to node $j \in \mathcal{N}'$ based on the smallest travel time $\min_{h\in\mathcal{H}} t_{klh}$ for each arc $(k,l) \in \mathcal{A}$. Then we can add the following constraints to improve the lower bounds:

$$\sum_{i\in\mathcal{N}_d} T_{ii} \geq \sum_{i,j\in\mathcal{N}'} x_{ij} S_{ij} + \sum_{i\in\mathcal{N}_c} s_i, \tag{32}$$

$$\sum_{i\in\mathcal{N}_d} T_{ii} \geq T_{jj} + \min_{d\in\mathcal{N}_d} S_{jd} \qquad \forall j \in \mathcal{N}_c, \tag{33}$$

$$T_{ii} \geq \sum_{p\in\mathcal{N}\backslash\mathcal{N}_o:(p,i)\in\mathcal{A}} y_{pii} \cdot \left\lceil \min_{\substack{k\in\mathcal{N}_c \\ l\in\mathcal{N}_o \\ m\in\mathcal{N}_d}} \{s_k + S_{lk} + S_{km}\} \right\rceil \qquad \forall i \in \mathcal{N}_d. \tag{34}$$

Constraint (32) ensures that the total of the arrival time of the vehicles back to the depot is at least the sum of all pairs of shortest paths $S_{ij},\ i,j \in \mathcal{N}'$, that are visited consecutively added to the total service time of the customers. Constraints (33) ensure that the total of the arrival time of the vehicles back to the depot is at least the departure time at customer node $j \in \mathcal{N}_c$ added to the shortest path from node j to the depot. Constraints (34) ensure that if a vehicle is used to visit customers, then its arrival time at the depot is at least the smallest sum over $k \in \mathcal{N}_c$ of the service time of customer k added to the shortest path from the depot to customer k and the shortest path from customer k to the depot.

We can improve shortest paths S_{ij} and end up with shortest paths P_{ij}, which are computed as follows. The shortest path P_{ij} from the depot $i \in \mathcal{N}_d$ to customer $j \in \mathcal{N}_c$ can be computed as $P_{ij} = S_{ij}$. Let e_i^1 be the earliest departure time at customer node $i \in \mathcal{N}_c$ computed as $e_i^1 = s_i + \min_{j \in \mathcal{N}_o} P_{ji}$ and let $\tilde{e}_i^1$ be the time period corresponding to e_i^1. Then, the shortest path P_{ij} from customer node $i \in \mathcal{N}_c$ to depot node $j \in \mathcal{N}_d$ can be computed as the shortest path based on the smallest travel time $\min\limits_{h \in \mathcal{H}: h \geq \tilde{e}_i^1} t_{klh}$ for each arc $(k,l) \in \mathcal{A}$. Let e_i^2 be the latest arrival time at customer node $i \in \mathcal{N}_c$ computed as $e_i^2 = H\overline{T} - s_i - \min_{j \in \mathcal{N}_d} P_{ij}$ and let $\tilde{e}_i^2$ be the time period corresponding to e_i^2. Then, the shortest path P_{ij} between two customers nodes $i \in \mathcal{N}_c$ and $j \in \mathcal{N}_c$ can be computed based on the smallest travel time $\min\limits_{h \in \mathcal{H}: \tilde{e}_i^1 \leq h \leq \tilde{e}_j^2} t_{klh}$ for each arc $(k,l) \in \mathcal{A}$.

We can then add the following valid inequalities to improve the lower bounds:

$$\sum_{i \in \mathcal{N}_d} T_{ii} \geq \sum_{i,j \in \mathcal{N}'} x_{ij} P_{ij} + \sum_{i \in \mathcal{N}_c} s_i, \tag{35}$$

$$\sum_{i \in \mathcal{N}_d} T_{ii} \geq T_{jj} + \min_{d \in \mathcal{N}_d} P_{jd} \qquad \forall j \in \mathcal{N}_c, \tag{36}$$

$$T_{ii} \geq \sum_{p \in \mathcal{N} \setminus \mathcal{N}_o : (p,i) \in \mathcal{A}} y_{pii} \cdot \left[\min_{\substack{k \in \mathcal{N}_c \\ l \in \mathcal{N}_o \\ m \in \mathcal{N}_d}} \{ s_k + P_{lk} + P_{km} \} \right] \qquad \forall i \in \mathcal{N}_d, \tag{37}$$

$$\delta_{llh} = 0 \qquad \forall h < \tilde{e}_l^1,\ l \in \mathcal{N}_c, \tag{38}$$

$$\delta^3_{ljkh} = 0 \qquad \forall l \in \mathcal{N}_c,\ h < \tilde{e}^1_l,\ j \in \mathcal{N} : (l,j) \in \mathcal{A},\ k \in \mathcal{N}', \tag{39}$$

$$T_{ll} \geq e^1_l \qquad \forall l \in \mathcal{N}_c, \tag{40}$$

$$\delta_{ilh} = 0 \qquad \forall l \in \mathcal{N}_c,\ i \in \mathcal{N}\backslash\{l\},\ h > \tilde{e}^2_l, \tag{41}$$

$$\delta^3_{ijlh} = 0 \qquad \forall l \in \mathcal{N}_c,\ h > \tilde{e}^2_l,\ i,j \in \mathcal{N} : (i,j) \in \mathcal{A}, \tag{42}$$

$$T_{il} \leq e^2_l \qquad \forall l \in \mathcal{N}_c,\ i \in \mathcal{N}\backslash\{l\}, \tag{43}$$

$$\delta^2_{pml} \leq e^2_l \qquad \forall l \in \mathcal{N}_c,\ (p,m) \in \mathcal{A} : m \neq l. \tag{44}$$

Constraints (35)–(37) are stronger versions of (32)–(34), by considering the shortest paths P_{ij} instead of S_{ij}. Constraints (38)–(40) arise from the fact that a vehicle cannot depart from customer node $l \in \mathcal{N}_c$ earlier than e^1_l, whereas constraints (41)–(44) arise from the fact that a vehicle cannot arrive at customer node $l \in \mathcal{N}_c$ later than e^2_l.

We will run our experiments on three different models. The basic model, which we call *Model 1*, corresponds to (1)–(31). It consists of the set of constraints that define the problem, i.e., (1)–(24), and the first set of valid inequalities that are imposed to strengthen the formulation, i.e., (25)–(31). The second model, which we call *Model 2*, is an extension of Model 1 by adding a new set of valid inequalities that are imposed to improve the lower bound, i.e., (32)–(34). Hence, Model 2 is composed of (1)–(34). The third model, which is called *Model 3*, is an extension of the basic model by adding another set of valid inequalities to improve the lower bound, i.e., (35)–(44). Thus, Model 3 is determined by (1)–(24) and (35)–(44).

3 Computational experiments

All experiments were performed on a desktop computer equipped with an Intel i7 processor and 64 GB of RAM, using CPLEX 12.6.3. In Section 3.1 the generation and the characteristics of the instances are described. In Section 3.2 we highlight the importance of implicitly considering all shortest paths between two consecutive nodes. Section 3.3 describes the generation of the initial solutions. The results of the instances, the impact of the different sets of valid inequalities and some insights on time-dependent optimization are discussed in Section 3.4.

3.1 Instances

Instances were generated from real data obtained from Québec City, Canada, and were created as follows. Several million GPS points were obtained from industrial partners operating in the Québec area. Each data point corresponds to the exact location of a truck, its speed and the time of the measurement. For each truck a measurement was taken at every few seconds, such that one is able to track its path as well the speed in each segment. By aggregating several thousand observations for each street segment, we were able to estimate the average speed on each street in any period of the day.

From this database, we selected representative areas of the city, discarded very small streets, and truncated the travel time information to correspond to business hours only. Following this procedure, we have generated 15 instances ranging from a very small area of the city containing 357 nodes and 616 arcs with only 10 customers, up to large instances containing 50 customers spread through a large area of the city containing 1612 nodes and 2810 arcs. The characteristics of the instances are summarized in Table 1. The complete set of instances can be found at `http://www.leandro-coelho.com/instances/time-dependent-vrp/`.

Table 1: Summary of the instances.

Group	# instances	# customers	# nodes	# arcs	# time periods
T	5	10 – 50	357	616	28
S	5	10 – 50	696	1226	52
M	5	10 – 50	1612	2810	52

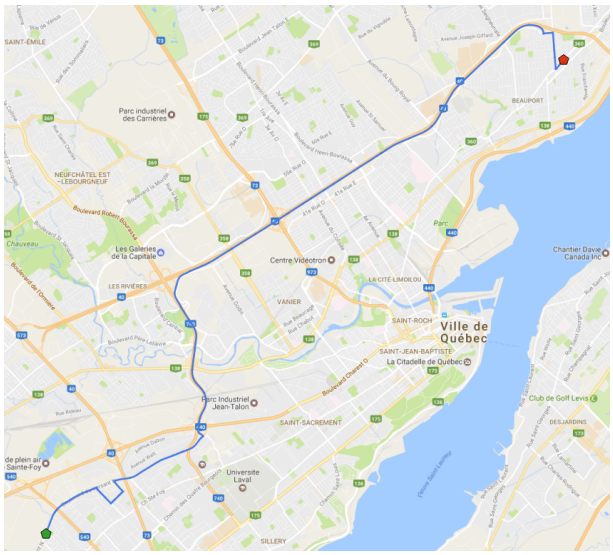
3.2 Importance of considering the underlying street network

We now show that the whole street network must be considered when computing the paths to be traveled between a pair of origin-destination nodes. We have selected a large instance and two nodes relatively far away from each other. We have then computed the time-dependent shortest path between these two nodes, varying the departure time. Figures 4 and 5 show eight different solutions obtained from this exercise. These figures show that not only the travel time varies significantly (from 1198 to 3014 seconds), the number of segments traveled also changes (from 93 to 115). A segment is defined as a small portion of the street, and varies in length depending on how it was coded in the geographical information system used. Typically a segment is delimited by the intersections of streets, but a long street may be composed of several smaller segments without intersections in between.

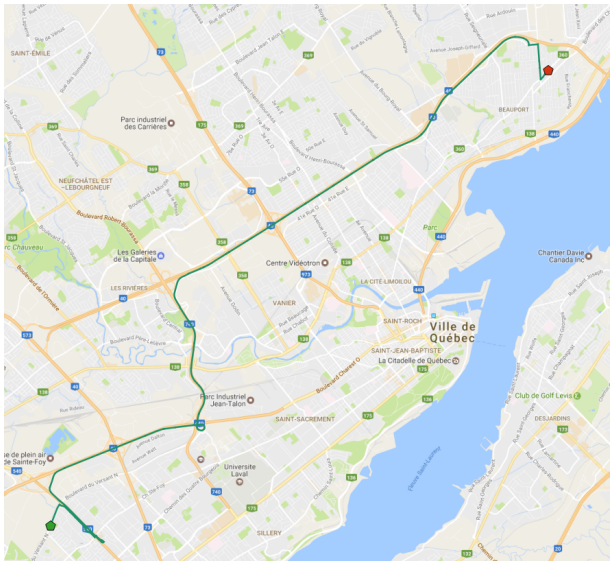
As can be seen from these maps, the shape of the solution changes considerably, with different highways and smaller road segments being used depending on the time of the day. We have observed that less than 4% of the street segments are common to all solutions, namely the access streets from the origin and to the destination.

3.3 Greedy initial solutions

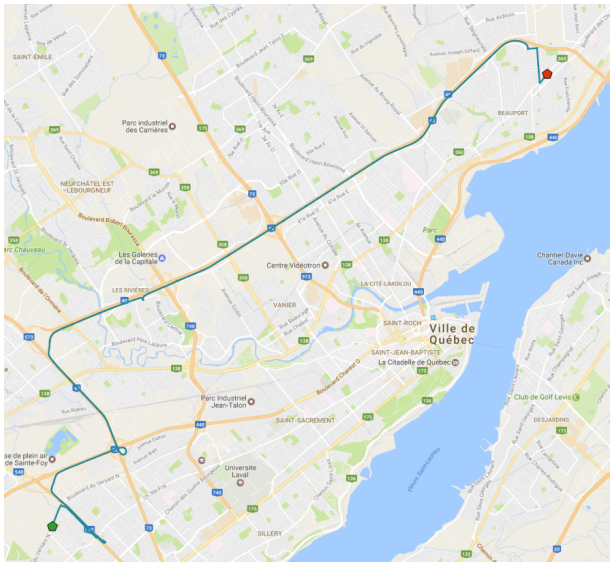
Initial solutions were computed by a simple heuristic based on the nearest neighbor idea. First, we have used a static travel time by taking for each arc its smallest traversal time (as done when computing the lower bounds). Then, starting from the depot, we extended



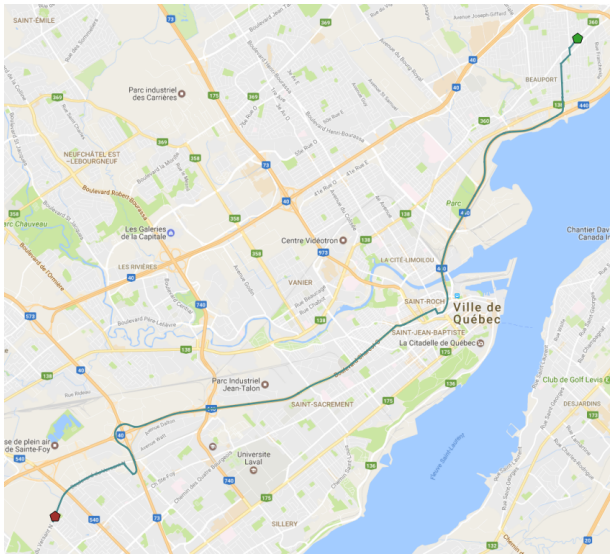
(a) Travel time: 1198 s, number of segments: 93



(b) Travel time: 2465 s, number of segments: 93

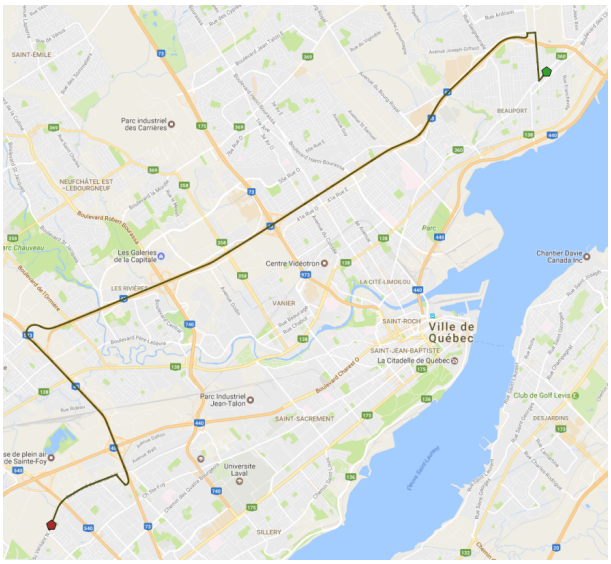


(c) Travel time: 3014 s, number of segments: 107

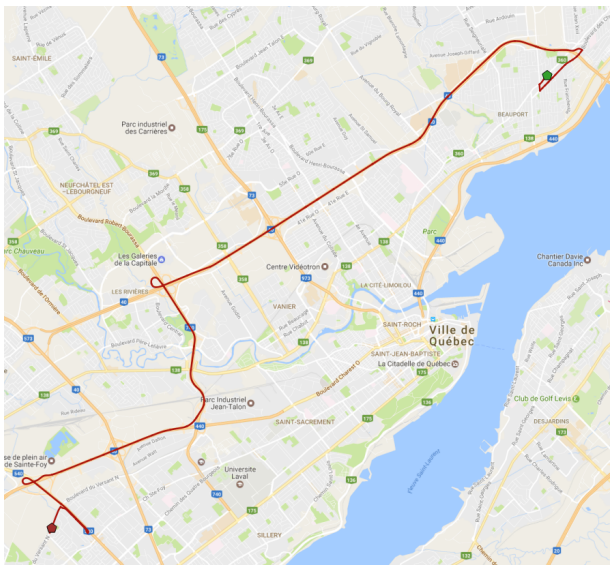


(d) Travel time: 1639 s, number of segments: 115

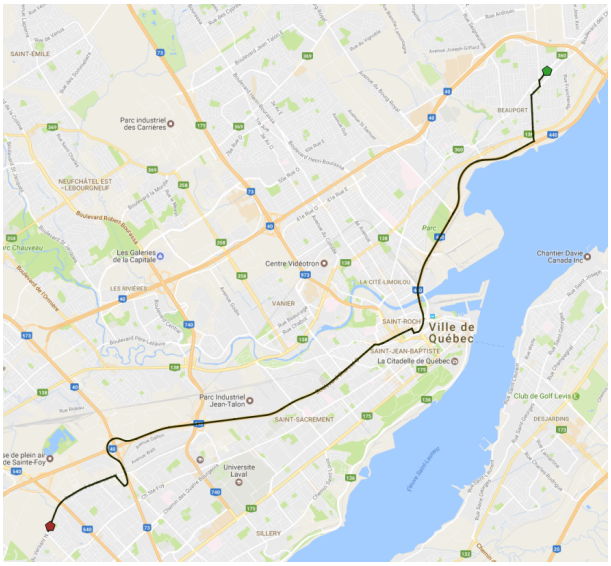
Figure 4: Optimal time-dependent travel times and number of segments for the same origin-destination, across different starting times



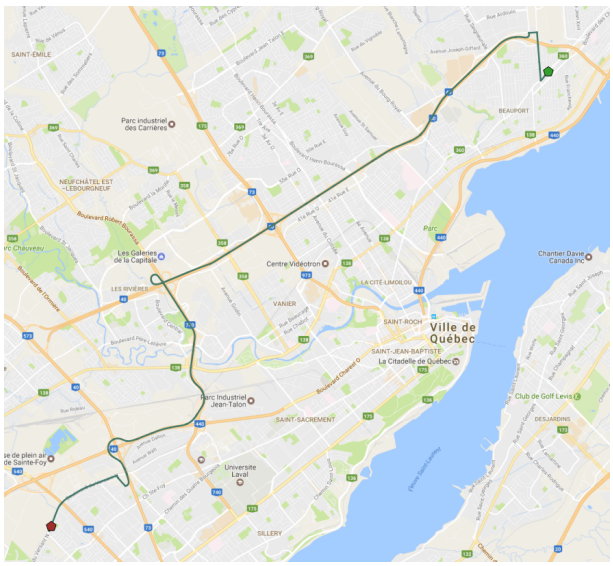
(a) Travel time: 1802 s, number of segments: 108



(b) Travel time: 2280 s, number of segments: 115



(c) Travel time: 1621 s, number of segments: 115



(d) Travel time: 1821 s, number of segments: 103

Figure 5: Optimal time-dependent travel times and number of segments for the same origin-destination, across different starting times

the timing labels on the network until the first customer was reached. At this point, we restarted the search until another customer was reached. Before incorporating the new customer to the route, some verification needed to be made. First and foremost, only non-visited customers were considered. Second, the load required by the new customer should be less than or equal to the available capacity in the truck. Finally, the time it would take to leave this customer and return to the depot should be within the length of the planning horizon. A posteriori, it is relatively simple to compute true traversal times and obtain the cost of the solution.

3.4 Computational results

For each instance and each model we set a time limit of 10 hours. The results in Table 2 show the impact of the different models on the number of instances solved and the problem size. The table reports for each instance group, #, the number of instances that did not encounter memory issues before starting to solve the problem. Moreover, for Model 1 it reports *Constr* and *Var*, the average number of constraints and variables over all instances in the group, and for Models 2 and 3 it reports Δ_C and Δ_V, the average increase in the number of, respectively, constraints and variables under Model i compared to Model 1 over all instances in the group. The increase in the number of constraints for one instance is computed as $(C_i - C_1)/C_1 \cdot 100$, where C_1 and C_i are the number of constraints under Models 1 and i, respectively. The increase in the number of variables is computed in a similar way. Instances M30, M40, and M50 are excluded from the results in Table 2, since these results were not available under all models, as some of them could not be built due to memory issues.

The results in Table 2 show that the number of constraints and variables in Model 2 do not differ significantly with Model 1. The number of instances that could be built without memory problems is, however, decreased under Model 2 from 14 to 12. Compared to Model 1, the size of the problem significantly decreases for Model 3, namely for the total of all instances the number of constraints and variables is decreased by 5.45% and

Table 2: Impact of the different models on the number of instances solved and the problem size.

	Model 1			Model 2			Model 3		
Group	#	Constr.	Var.	#	Δ_C	Δ_V	#	Δ_C	Δ_V
T	5	1,331,382	946,945	5	0.00	0.00	5	–6.70	–6.14
S	5	4,414,168	3,146,947	5	0.00	0.00	5	–4.35	–4.00
M	4	5,167,447	3,645,341	2	0.00	0.00	5	–4.49	–4.13
Total	14	3,255,220	2,313,345	12	0.00	0.00	15	–5.45	–5.01

5.01%, respectively. Moreover, all 15 instances could be built without memory problems, which may be explained by a reduction in the number of constraints and variables. An explanation of the reduction in the number of constraints and variables in Model 3 is the introduction of constraints (38)–(44). Comparing Models 2 and 3, constraints (35)–(37) in Model 3 are stronger versions of (32)–(34) from Model 2 using the shortest paths P_{ij} instead of S_{ij}. Moreover, in Model 3 we have added constraints (38)–(44) that are derived from the generation of the shortest paths P_{ij}. Thus, where Model 3 uses stronger versions of the shortest paths, it also introduces completely new constraints that can be seen as by-products from the generation of the shortest paths P_{ij}, which help reduce the problem size.

Detailed computational results are given in Table 3, where we report *Inst*, the name of the instance, *Init*, the objective value of the initial solution, and for Models 1–3, *Constr*, the number of constraints of the reduced MIP, *Var*, the number of variables of the reduced MIP, $\underline{z}$, the value of the lower bound, and $\overline{z}$, the value of the upper bound. Note that there are no results for instance $M30$ under Model 1 and for instances $M30$, $M40$ and $M50$ under Model 2, since the program could not built these instances due to a lack of memory. Note that our emphasis lies in improving the lower bound, as these are the differentiating aspects of our models. We understand that MIPs this big cannot be efficiently solved by branch-and-bound, and that several simple heuristics should be able to outperform the upper bounds we provide. However, our lower bounds are expected to be tight, as we discuss in detail in this section.

Table 3: Detailed computational results.

		Model 1				Model 2				Model 3			
Inst.	Init.	Constr.	Var.	$\underline{z}$	$\overline{z}$	Constr.	Var.	$\underline{z}$	$\overline{z}$	Constr.	Var.	$\underline{z}$	$\overline{z}$
T10	31,178	460,985	326,042	4.00	32,259	461,002	326,045	26,992.17	48,159	425,090	302,310	27,400.20	34,112
T20	50,499	882,659	625,881	3.80	70,793	882,694	625,891	45,402.40	53,600	818,586	583,896	45,601.18	71,258
T30	85,328	1,328,727	944,586	0.00	87,477	1,328,762	944,584	76,637.09	85,784	1,225,682	876,630	76,871.06	89,444
T40	77,768	1,763,342	1,254,968	3.80	96,422	1,763,384	1,254,962	70,948.36	96,434	1,665,284	1,192,400	71,082.94	96,080
T50	90,103	2,221,197	1,583,249	3.80	93,571	425,085	302,292	81,499.30	95,430	2,107,374	1,511,475	81,693.62	95,735
S10	33,686	1,528,155	1,086,759	0.00	45,374	1,528,167	1,086,759	28,775.00	45,977	1,454,351	1,037,859	29,090.62	45,179
S20	59,184	2,976,244	2,118,379	8.33	–	2,976,272	2,118,384	52,111.00	–	2,837,710	2,027,021	52,366.67	–
S30	92,260	4,474,712	3,188,980	0.00	92,260	4,474,749	3,188,982	83,234.94	92,260	4,251,587	3,040,966	83,592.87	92,260
S40	80,334	5,755,163	4,104,691	0.00	–	5,755,206	4,104,692	71,822.03	93,471	5,556,509	3,977,702	72,027.79	–
S50	111,505	7,336,565	5,235,927	0.00	–	7,336,622	5,235,928	99,448.30	–	7,056,984	5,055,052	99,663.41	139,400
M10	33,582	3,525,741	2,486,087	8.33	–	3,525,755	2,486,088	29,407.37*	46,502*	3,342,230	2,365,568	29,721.35	45,071
M20	59,528	6,809,153	4,804,595	0.00	–	6,809,168	4,804,587	52,775.59*	69,938*	6,470,645	4,583,179	53,221.53*	69,170*
M30	93,993									9,600,322	6,803,353	84,736.60	–
M40	85,707	13,217,576	9,335,844	0.00	87,235					12,735,589	9,029,734	73,499.44	91,476
M50	112,891	16,348,691	11,556,167	0.00	–					15,698,811	11,139,733	101,232.01	140,216

* indicates that the results are truncated, because the execution was aborted due to a lack of memory.

Table 4 reports for Models 1–3, $\underline{z}$, the value of the lower bound, and *Gap (%)*, the gap between the lower bound and the best found solution, computed as $(\underline{z} - z_{best})/z_{best} \cdot 100$, where z_{best} is the best obtained solution for the given instance.

Table 4: Impact of the different models on the lower bounds.

	Model 1		Model 2		Model 3	
Inst	$\underline{z}$	Gap (%)	$\underline{z}$	Gap (%)	$\underline{z}$	Gap (%)
T10	4.00	–99.99	26,992.17	–13.43	27,400.20	–12.12
T20	3.80	–99.99	45,402.40	–10.09	45,601.18	–9.70
T30	0.00	–100.00	76,637.09	–10.19	76,871.06	–9.91
T40	3.80	–100.00	70,948.36	–8.77	71,082.94	–8.60
T50	3.80	–100.00	81,499.30	–9.55	81,693.62	–9.33
S10	0.00	–100.00	28,775.00	–14.58	29,090.62	–13.64
S20	8.33	–99.99	52,111.00	–11.95	52,366.67	–11.52
S30	0.00	–100.00	83,234.94	–9.78	83,592.87	–9.39
S40	0.00	–100.00	71,822.03	–10.60	72,027.79	–10.34
S50	0.00	–100.00	99,448.30	–10.81	99,663.41	–10.62
M10	8.33	–99.98	29,407.37	–12.43	29,721.35	–11.50
M20	0.00	–100.00	52,775.59	–11.34	53,221.53	–10.59
M30					84,736.60	–9.85
M40	0.00	–100.00			73,499.44	–14.24
M50	0.00	–100.00			101,232.01	–10.33
Average		–100.00		–11.13		–10.78

Results in Table 4 show that the basic model provides poor lower bounds. For this model, the gaps are between –99.98% and –100.00% and for 8 instances the lower bounds remain at 0.00. The results also show that the valid inequalities imposed to improve the lower bounds do have a large impact. For Model 2, in which a set of valid inequalities are added to the basic model, the gaps between the best found solutions and the lower bounds are improved significantly and the gaps with the best found solutions are between –8.77% and –14.58%. The valid inequalities added to the basic model in Model 3 are even stronger than the set of valid inequalities in Model 2. The lower bounds obtained by Model 3 are on average 0.59% higher than the lower bounds obtained by Model 2, where the instances $M30$, $M40$ and $M50$ are not included in the calculations. Under Model 3 the gaps are

between –8.60% and –14.24%.

In Table 5 we give some insights on time-dependent optimization by showing the impact of traffic. We have considered two situations: in the first one we ignore traffic and assume that one can drive as close as possible to the nominal speed, i.e., we have taken the best traversal time for each arc, and in the second situation we are as pessimistic as possible, i.e., we have taken the worst traversal time for each arc. We evaluated the initial TDSPVRP solutions on the graphs with the best and worst traversal times. Table 5 reports *Init*, the initial solution evaluated on the TDSPVRP graph, and for the minimum and maximum traversal time it reports *Obj*, the objective of the initial solution evaluated on the graphs with the minimum and maximum traversal time, respectively, and *Gap (%)*, the gap between *Obj* and *Init* computed as $(Obj - Init)/Init \cdot 100$.

Table 5: Impact of traffic.

		Min. traversal time		Max. traversal time	
Inst.	Init.	Obj.	Gap (%)	Obj.	Gap (%)
T10	31,178	27,369	–12.22	36,570	17.29
T20	50,499	47,109	–6.71	–	
T30	85,328	77,797	–8.83	91,984	7.80
T40	77,768	73,020	–6.11	–	
T50	90,103	83,563	–7.26	–	
S10	33,686	29,348	–12.88	42,143	25.11
S20	59,184	53,040	–10.38	–	
S30	92,260	84,033	–8.92	–	
S40	80,334	72,879	–9.28	–	
S50	111,505	101,649	–8.84	–	
M10	33,582	29,773	–11.34	41,451	23.43
M20	59,528	54,539	–8.38	–	
M30	93,993	86,912	–7.53	–	
M40	85,707	75,189	–12.27	–	
M50	112,891	102,958	–8.80	–	
Average			–9.32		18.41

The average gap between the initial solutions evaluated on the graph with the minimum traversal time and the initial solutions evaluated on the TDSPVRP graph is –9.32%.

Hence, when we do not take into account traffic and delays, the sum of the arrival times back at the depot is underestimated with 9.32% on average. Note that companies that do not take any information on traffic into account will perform even worse, as we did consider *some* real information by using the best traversal time as observed from the database, which can still be worse than the nominal speed. Neglecting traffic will result in substantial delays at the locations that need to be visited, which in turn would require more vehicles and more mileage to perform the deliveries. Comparing the initial solutions evaluated on the graph with the maximum traversal times with the solutions evaluated on the TDSPVRP graph, we see that most of the solutions that are feasible in the TDSPVRP graph are no longer feasible under the graph based on the maximum traversal time, since they exceed the time horizon. Hence, using the graph with maximum traversal times eliminates solutions that are feasible under the TDSPVRP and are not necessarily sub-optimal. To conclude, results show that the impact of traffic is large and it is important to incorporate it into our routing models.

4 Conclusions

In this paper, we have introduced the time-dependent shortest path and vehicle routing problem. We provided a mathematical formulation for the problem and we developed valid inequalities to strengthen this formulation and to improve the lower bounds. We have developed three models, differing by a set of valid inequalities developed to improve the lower bounds, based on different and more intricate shortest paths. We have created a set of 15 instances generated from real traffic data on the road network in Québec City, Canada. The instances range from 357 nodes, 616 arcs, 28 time periods and 10 customers up to 1612 nodes, 2810 arcs, 52 time periods and 50 customers. These instances can serve as benchmark instances for the TDSPVRP. We provided initial solutions by using a simple heuristic based on the nearest neighbor idea. Results have shown that the problem size of the instances under Models 1 and 2 are almost the same, whereas the number of

constraints and variables is on average reduced by 5.45% and 5.01%, respectively, under Model 3. Moreover, under Model 3, less instances encountered memory issues before starting to solve the problem. The standard formulation of Model 1 provides very weak lower bounds, where the gap with the best found solutions is between –99.98% and –100.00%. The valid inequalities that are imposed to improve the lower bounds are strong and significantly improve the gaps, i.e., the gaps between the lower bound and the best found solution are between –8.77% and –14.58% for Model 2 and between –8.60% and –14.25% for Model 3. The use of the stronger set of valid inequalities to improve the lower bounds, based on different shortest paths, provides lower bounds that are on average 0.59% higher.

We provided a sensitivity analysis that supports the importance of including traffic in our routing models, by showing that ignoring it can impose substantial delays. To find good upper bounds for the TDSPVRP, state-of-the-art heuristics and more elaborated exact methods should be developed in future research. The benchmark instances generated and the lower bounds provided in this research can be used to test the results of those heuristics.

References

[1] T. Bektaş and G. Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011.

[2] I. Chabini. Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time. *Transportation Research Record: Journal of the Transportation Research Board*, 1645:170–175, 1998.

[3] H.-K. Chen, C.-F. Hsueh, and M.-S. Chang. The real-time time-dependent vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 42(5):383–408, 2006.

[4] K. L. Cooke and E. Halsey. The shortest route through a network with time-dependent internodal transit times. *Journal of Mathematical Analysis and Applications*, 14(3):493–498, 1966.

[5] B. Ding, J.X. Yu, and L. Qin. Finding time-dependent shortest paths over large graphs. *Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology*, pages 205–216, 2008.

[6] A. V. Donati, R. Montemanni, N. Casagrande, A.E. Rizzoli, and L. M. Gambardella. Time dependent vehicle routing problem with a multi ant colony system. *European Journal of Operational Research*, 185(3):1174–1191, 2008.

[7] S. E. Dreyfus. An appraisal of some shortest-path algorithms. *Operations Research*, 17(3):395–412, 1969.

[8] A. Franceschetti, D. Honhon, T. Van Woensel, T. Bektaş, and G. Laporte. The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological*, 56:265–293, 2013.

[9] H. Hashimoto, M. Yagiura, and T. Ibaraki. An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete Optimization*, 5(2):434–456, 2008.

[10] Y. Huang, L. Zhao, T. Van Woensel, and J.-P. Gross. The time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological*, 95:169–195, 2017.

[11] I. Kara, G. Laporte, and T. Bektaş. A note on the lifted Miller-Tucker-Zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operational Research*, 158(3):793–795, 2004.

[12] Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte. The fleet size and mix pollution-routing problem. *Transportation Research Part B: Methodological*, 70:239–254, 2014.

[13] Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte. The impact of depot location, fleet composition and routing on emissions in city logistics. *Transportation Research Part B: Methodological*, 84:81–102, 2016.

[14] A. L. Kok, E. W. Hans, and J. M. J. Schutten. Vehicle routing under time-dependent travel times: the impact of congestion avoidance. *Computers & Operations Research*, 39(5):910–918, 2012.

[15] C. Malandraki and M. S. Daskin. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation Science*, 26(3):185–200, 1992.

[16] A. Orda and R. Rom. Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the ACM*, 37(3):607–625, 1990.

[17] D. Soler, J. Albiach, and E. Martínez. A way to optimally solve a time-dependent vehicle routing problem with time windows. *Operations Research Letters*, 37(1): 37–42, 2009. ISSN 0167-6377.

[18] A. K. Ziliaskopoulos and H. S. Mahmassani. Time-dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications. *Transportation Research Record*, pages 94–94, 1993.