Publié par :
Published by:
Publicación de la:

Faculté des sciences de l'administration
2325, rue de la Terrasse
Pavillon Palasis-Prince, Université Laval
Québec (Québec) Canada G1V 0A6
Tél. Ph. Tel. : (418) 656-3644
Télec. Fax : (418) 656-7047

Disponible sur Internet :
Available on Internet
Disponible por Internet :

http://www4.fsa.ulaval.ca/la-recherche/publications/documents-de-travail/

DOCUMENT DE TRAVAIL 2019-001

Determining Time-Dependent Minimum Cost Paths under Several Objectives

Hamza HENI
Leandro C. COELHO
Jacques RENAUD

Document de travail également publié par le Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, sous le numéro CIRRELT-2019-01

Janvier 2019

Dépôt legal – Bibliothèque et Archives nationales du Québec, 2019
Bibliothèque et Archives Canada, 2019

ISBN 978-2-89524-480-6 (PDF)

Determining Time-Dependent Minimum Cost Paths under Several Objectives[+]

Hamza Heni, Leandro C. Coelho*, Jacques Renaud

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

**Corresponding author: leandro.coelho@cirrelt.ca*

ABSTRACT

As the largest contributor to greenhouse gas (GHG) emissions in the transportation sector, road freight transportation is the focus of numerous strategies to tackle increased pollution. One way to reduce emissions is to consider congestion and being able to route traffic around it. In this paper we study time-dependent minimum cost paths under several objectives (TDMCP-SO), in which the objective function comprises GHG emissions, driver and congestion costs. Travel costs are impacted by traffic due to changing congestion levels depending on the time of the day, vehicle types and carried load. We also develop time-dependent lower and upper bounds, which are both accurate and fast to compute. Computational experiments are performed on real-life instances that incorporate the variation of traffic throughout the day, by adapting Dijkstra's label-setting algorithm according to different cost computation methods. We show that explicitly considering first-in, first-out (FIFO) consistency using time-varying speeds allows the efficient computation of tight time-dependent bounds. Our computational results demonstrate that the TDMCP-SO is more difficult to solve to optimality but the proposed algorithms is shown to be robust and efficient in reducing the total cost even for large instances in an environment of varying speeds, outperforming those based on the link travel time model and on the smoothing method according to each optimization objective, flexible departure times, and different load patterns.

Keywords: Time-dependent networks, congestion, emission, quickest path, bounds, label-setting algorithm, link travel time, flow speed model.

Acknowledgments: This research was partly supported by grants 2014-05764 and 0172633 from the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Centre d'Innovation en Logistique et Chaîne d'Approvisionnement Durable (CILCAD) that receives financial support from the Green Fund under priority 15.2 of the 2013-2020 Action Plan on Climate Change, a priority implemented by Transition énergétique Québec (TEQ). These supports are gratefully acknowledged. We also thank Mr Jean-Philippe Gagliardi, President of Logix Operations inc, for providing us with real data from an important wholesaler partner in Québec City.

[+] Revised version of 2017-014 (CIRRELT 2017-62)

1 Introduction

Road freight transportation is a significant contributor to greenhouse gas (GHG) emissions [15]. This is mostly driven by increased traffic congestion due to the high number of vehicles on urban areas. When traveling in cities, fuel consumption and GHG emissions are highly affected by speed levels depending on the paths used by vehicles. To reduce the emissions intensity and environmental pollution caused by road freight transportation activities, new alternative planning and coordination strategies directly related to routing and scheduling operations are required for both operational and environmental considerations [38, 23].

Most of the existing research assumes that trucks can travel at the emissions-minimizing speed, which largely ignores the effect of congestion, notably in urban areas. However, few works have demonstrated the importance of speed in minimizing emissions and travel costs [18, 30, 20]. In this work we study a variant of the minimum-cost path across time-dependent networks which we call the time-dependent minimum cost path under several objectives (TDMCP-SO). Here, the cost of an arc depends not only on distance but also on fuel consumption (the rate of GHG emissions) and on driver costs, which are all affected by speed variation. The aim is to determine the least cost path based on a time-dependent network with time-varying speeds, which is more challenging than when using a traditional objective due to the nonlinear relationship between speed and fuel consumption. To the best of our knowledge, there is a lack of theoretical and computational experiments in terms of lower and upper bounds to the TDMCP-SO and benchmarks regarding different cost calculation methods in an environment of varying speeds and loads. Doing so, with data obtained for Québec City, we test our algorithms on a large road network with real traffic data. We adapt Dijkstra's label-setting algorithm to account for time-dependent traffic and for our comprehensive objective function composed of GHG emissions, fuel, and driver costs to efficiently compute lower and upper bounds on the overall cost.

By conducting extensive experiments on the road network of Québec City using a dataset of millions of speed observations, our results show that the the fast computation of point-to-point least cost paths through our first-in, first-out (FIFO) consistent method outperforms those based on the link travel time model (LTM) [36] and on the smoothing method [21] in terms of travel time, fuel consumption, and cost savings as well as computational time.

This paper makes several important contributions to the literature:

- we prove that, under the FIFO property, the least cost path obtained by ignoring traffic congestion can be no worse than the optimal path cost according to the heaviest congestion factor applied to all arcs at each time interval;
- we propose an efficient method to obtain tight time-dependent bounds, reducing the computational burden, and investigate when it is important to incorporate the load carried by the vehicle and traffic congestion into the lower and upper bounding algorithms;
- we propose an effective approach for computing travel cost and GHG emissions in time-dependent networks under the FIFO consistency overcoming the challenge of the non-linearities of emissions. This ensures that our solution methods applied to solve the TDMCP-SO account for the impact of speed variations on the optimization of a chosen path;
- finally, we provide an extensive set of benchmarks showing the effectiveness of our FIFO-consistent method compared to LTM and smoothing method providing the best solutions for each optimization criterion and coherent results under flexible departure times.

The remainder of the paper is organized as follows. Section 2 provides a literature review of the TDQPP and closely related problems. In Section 3, we provide a formal description of the TDMCP-SO, and present some of its properties. Section 4 describes the proposed lower and upper bounds on the cost. Section 5 is devoted to extending the dynamic Dijkstra's label-setting algorithm, incorporating our lower and upper bounds on that algorithm. In Section 6, we give details on the benchmarks created from the Québec metropolitan area and validate the performance of our algorithms providing a detailed experimental evaluation of the proposed FIFO-consistent method and the designed bounds. Our conclusions are presented in Section 7.

2 Literature review

The TDMCP-SO is a problem in the field of green road freight transportation [15], and more specifically close to the pollution-routing problem (PRP) [3]. A number of recent contributions on the PRP have addressed both operational and GHG emissions-related objectives [14, 23].

Most of these contributions consider the shortest path between each pair of customers as fixed. Time-dependent shortest path problems (TDSPPs) have been studied in most cases in the context of other objectives, such as determining Quickest Path (QP) [12, 6], time-dependent least emissions

path (TDLEP) [18, 19] and minimum-cost path (MCP) [16]. The TDMCP-SO is an extension of the TDQPP considering a time-dependent travel cost. It can be seen as a variant of the MCP over time-dependent networks, which is $\mathcal{NP}$-hard as stated in Dean [10] and demonstrated by Di Bartolomeo et al. [16] and Wen et al. [46]. Since the TDMCP-SO is a variant of the MCP which aims to find the least travel cost path over time-dependent networks considering a cost function encompassing GHG emissions and driver costs, it is then also $\mathcal{NP}$-hard. Polynomial-time algorithms for the TDQPP can be adapted to find the MCP [10, 5, 11].

In what follows, we review contributions on the TDQPP in Section 2.1 and on the time-dependent emissions-minimizing path problem in Section 2.2.

2.1 The time-dependent quickest path problem

A large part of the literature dealing with shortest path on time-dependent networks aims at finding a path with the least travel time, also known as the TDQPP. This problem has been first introduced by [7]. The classical Dijkstra's label-setting algorithm can be used to determine quickest paths in time-dependent networks, in which the FIFO property was implicitly considered as it is consistent with the requirements imposed by real transportation networks. Under FIFO consistency the TDQPP can be solved optimally and efficiently in polynomial time by adapting any label-setting shortest path algorithm [9].

Moreover, many existing works have not explicitly considered whether the FIFO property holds [34, 18], requiring additional steps to manage cycles in non-FIFO networks even when waiting is not permitted. Sherali et al. [39] show that a network with a single non-FIFO arc yields a TDQPP algorithm which can no longer be solved in polynomial time.

With the aim of ensuring that the time-dependent network is consistent with the FIFO consistency, Sung et al. [41] proposed the flow speed model (FSM) in which the flow speed of each portion of an arc depends on the time it is traversed. They developed a solution method based on Dijkstra's label-setting algorithm and showed that the computation of an optimal solution using the FSM is easier than the one based on the link travel time model (LTM). In fact, the LTM does not guarantee the FIFO property as the arc travel time is fixed by the time at which the arc is entered. The determination of quickest paths with LTM requires some additional steps to ensure the FIFO consistency by eliminating potential cycles if waiting at nodes is not allowed [34] or deriving a smoothed travel time function (STTF) [21]. Recently, Yang and Zhou [47] proposed a branch-and-

bound method to solve the TDQPP in a space-time network by defining time-dependent nodes based on the departure and arrival times at each physical node.

Ichoua et al. [31] proposed a time-dependent speed model that respects the FIFO consistency. The main point of their model is that the speed of each arc depends on the period. Hence, the speed across each arc changes when the boundary between two consecutive time intervals is crossed. Later, Van Woensel et al. [44] proposed a queuing approach to capture traffic congestion and model travel times. A study by Kok et al. [33] proposed a speed model that satisfies the FIFO property to reflect traffic congestion in real road networks.

A best-first search heuristic for the fast computation of quickest path in a time-dependent network is the A* goal-directed search algorithm [27]. A* can be seen as an efficient adaptation of Dijkstra's algorithm that determines the quickest path on time-dependent networks using time-to-destination lower bounds satisfying the FIFO property.

In their work Gao and Chabini [24] studied optimal routing policies to find the quickest path in stochastic time-dependent networks, where arc travel times are modeled as random variables with time-dependent distributions. Gao and Huang [25] consider real-time travel information in the analysis of the optimal adaptive quickest path in stochastic time dependent flow-independent networks. A decreasing order of time with partitions heuristic algorithm based on Gao and Chabini [24] has been applied to find the optimal solution. Similarly, Huang and Gao [29] designed an exact label-correcting complete dependency-path algorithm to find the least time shortest path. To handle spatial travel time correlations related to an expected least time path in time-dependent and stochastic network, Yang and Zhou [47] applied a Lagrangian relaxation-based solution approach. Sun et al. [40] studied time-dependent traffic graphs, generated from historical traffic data to predict the traveling time and to dynamically find the quickest path for drivers. Solution methods for determining the expected quickest paths are crucial for the coordination and the planning of routing and could be enhanced with the availability of real-time traffic information.

Ghiani and Guerriero [26] proposed an effective lower bound for the quickest path problem, which was embedded into an A* algorithm. Calogiuri et al. [6] studied the properties and bounds of TDQPP. Using the time-dependent speed model of Ichoua et al. [31], they prove that under the FIFO consistency, if the congestion factors of all links are set to the lightest congestion factor, the TDQPP can be solved as a QPP with suitable-defined fixed travel times. We extend this development to the context of our paper.

2.2 Time-dependent pollution-routing and emissions-minimized paths

Demir et al. [15] provided a review of several fuel consumption models including the Methodology for Calculating Transportation Emissions and Energy Consumption (MEET) developed by Hickman et al. [28] and the Comprehensive Modal Emissions Model (CMEM) designed by Barth and Boriboonsomsin [1, 2]. The CMEM considers the impact of vehicle load on fuel consumption, while the MEET uses a load correction factor to take the vehicle load into account when computing fuel consumption.

Bektaş and Laporte [3] introduced the pollution-routing problem. Based on the CMEM, they minimize GHG emissions by determining the optimal speed with respect to the load carried by the vehicle, fuel consumption and driver induced costs. Later, Demir et al. [13] extend it by applying a speed optimization algorithm, identifying the optimal speed on each arc in order to minimize the expected costs of fuel consumption and driver wages.

Jabali et al. [32] studied the Emission Vehicle Routing Problem (EVRP), which uses the MEET to derive the GHG emissions. However, this study ignores the load when calculating the emissions. Focusing on the analysis of time-dependent costs as a function of speed, load and fuel consumption, Franceschetti et al. [22] extended the PRP to a time-dependent setting using the time-dependent travel time model of Jabali et al. [32]. Recently, Franceschetti et al. [23] developed a metaheuristic approach to solve the PRP under congestion, which integrates departure time and speed optimization procedures.

In Wen and Eglese [45], the authors solve the vehicle routing problem (VRP) with time-dependent speeds, where the total cost involves fuel cost, driver cost and congestion charge. Their model is based on MEET and the impact of vehicle load is not considered. In their work, fixed congestion charges are applied once per day for each vehicle.

The results of the previous works show that the traditional objectives consisting of minimizing travel times do not necessarily imply the minimization of either fuel or driver costs, and that least cost solutions do not imply an GHG emissions-optimal solution. Few papers have addressed path flexibility and GHG emissions-minimized paths. The exceptions are the works of Wen et al. [46], Qian and Eglese [37], Ehmke et al. [18, 19] and Huang et al. [30]. Recently, Ehmke et al. [20] addressed the impact of minimizing combined fuel and driver costs on path choices introducing non-linearity challenges not found in traditional routing objectives. Their results show the value of

comprehensively modeling total cost. Following these works, we assume that vehicles must travel at the speed of traffic and do not have the ability to control their speed, which requires controlling the paths of vehicles in a way that minimizes costs. Additionally, we consider that in time-dependent networks, congestion is variable across each segment depending on the corresponding period. Indeed, there is a gap in the PRP and TDMCP research related to the integration of non-linear GHG emissions models into lower and upper bound methods for the TDMCP-SO. Thus, to build on existing literature and to avoid redundancy with their key findings, in this work we focus on both computing tight and accurate time-dependent bounds for the TDMCP-SO on large road network. We also study the computational efficiency and the solution quality of our FIFO-consistent method in an environment of varying speeds and loads through an extensive comparative analysis with cost computation methods based on LTM and LTM with STTF (LTM-STTF).

3 Formal description and problem statement

In this section, we introduce our notation, give a formal definition for the TDMCP-SO and describe some of its properties. Let $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{Z}, \mathcal{S})$ be a directed time-dependent network, where $\mathcal{V}$ is the set of nodes, and $\mathcal{A} \subseteq \{(i,j) \in \mathcal{V} \times \mathcal{V}, i \neq j\}$ is a set of arcs. The number of nodes and arcs are $|\mathcal{V}| = n$ and $|\mathcal{A}| = m$. We assume that $\mathcal{G}$ is strongly connected, thus, there is a path from every node to all other nodes. The time-dependent network is considered at a set of discrete times $\mathcal{Z} = \{t_0, t_0 + \delta ..., t_0 + H\delta\}$, with $\delta > 0$ being the smallest increment of time over which a change in the congestion pattern occurs. The time horizon $\mathcal{T}$ is divided into H time slots $Z_h = [z_h, z_{h+1}[$, such that $z_h = t_0 + h\delta$, where $h = 0, 1, 2, ..., H-1$. Let $\mathcal{S} = \{s_{ij}^h\}$ represent the set of time-dependent arc travel speeds, where for each arc $(i,j) \in \mathcal{A}$ s_{ij}^h represents the travel speed value during the time slot Z_h. Time-dependent travel times as well as costs vary for each departure time $t \in \mathcal{T}$. With each arc (i,j) are associated two time-dependent functions $\tau : \mathcal{A} \longrightarrow \mathbb{R}^+$ and $c : \mathcal{A} \longrightarrow \mathbb{R}^+$ which assign, respectively, travel time $\tau_{ij}(t)$ and travel cost $c_{ij}(t)$ related to the time at which a vehicle leaves node i. Travel time functions $\tau_{ij}(t)$ are piecewise linear and satisfy the FIFO property.

The speed at which a vehicle travels on arc (i,j) is constrained by a lower bound and an upper bound, denoted $\mathcal{L}_{ij}$ and $\mathcal{U}_{ij}$, respectively, usually imposed by traffic. A unit of GHG emitted (usually in kilograms) has an estimated cost c_e.

Given a starting time t the TDMCP-SO aims to determine a path $p = (o = v_0, ..., v_i, ..., v_j, ..., v_k =$

d) such that the time-dependent total cost $\varphi_p(t)$ between source $o \in \mathcal{V}$ and destination $d \in \mathcal{V}$ is minimum.

Following the FSM, when a vehicle traverses arc (i, j) of length L_{ij}, speed may change when the boundary between two consecutive time slots is crossed before reaching j. Since arc travel time is then obtained by summing up the travel times used for each section traversed at different speeds, FSM ensures FIFO consistency by explicitly considering the variability in speed on each arc at different time intervals. Hence, with FSM we assume that speed s_{ij}^h on arc (i, j) depends on the time interval:

$$s_{ij}^h = \sigma_{ijh} u_{ij}, \tag{1}$$

where $\sigma_{ijh} \in [0, 1]$ represents the congestion ratio of arc (i, j) in the time interval Z_h, and u_{ij} is the maximum speed of arc $(i, j) \in \mathcal{A}$ during the horizon $\mathcal{T}$.

For a given arc (i, j) let l_{ij}^h denote the portion of the length L_{ij} traveled during time slot Z_h. Let h_t and h_γ be the indices of time slots where the start time $\gamma_i^p(t)$ at node i and the arrival time $\gamma_j^p(t)$ at node j belong to, respectively, with $h_t \in \{0, ..., H-1\}$ and $h_\gamma \in \{h_t, ..., H-1\}$. The travel time along an arc is, *in the worst situation,* the sum of three portions of time (see Figure 1):

(i) The time associated with the first interval h_t when entering the arc: $z_{h_t+1} - \gamma_i^p(t)$.

(ii) The duration of the $(h_\gamma - h_t - 1)$ intermediate time intervals crossed when travelling along arc (i, j): $\sum_{h=h_t+1}^{h_\gamma-1} (z_{h+1} - z_h)$, where $h = h_t + 1, ..., h_\gamma - 1$.

(iii) The time related to the last time slot when leaving the arc: $\gamma_j^p(t) - z_{h_\gamma}$.

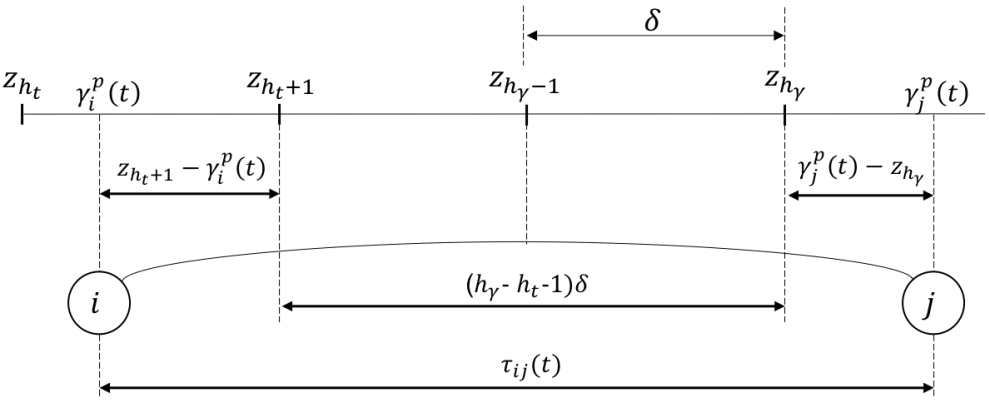

Figure 1: Illustration of travel time computation

Therefore, we can express the travel time of each arc as follows:

$$\tau_{ij}(\gamma_i^p(t)) = \begin{cases} L_{ij}/(\sigma_{ijh_t} u_{ij}) & \text{if } h_\gamma = h_t \\ \gamma_j^p(t) - \gamma_i^p(t) & \text{if } h_\gamma = h_t + 1 \\ \left(z_{h_t+1} - \gamma_i^p(t)\right) + (h_\gamma - h_t - 1)\,\delta + \left(\gamma_j^p(t) - z_{h_\gamma}\right) & \text{if } h_\gamma > h_t + 1. \end{cases} \tag{2}$$

The arrival time γ_j^p at node j is expressed as follows:

$$\gamma_j^p(t) = \begin{cases} L_{ij}/s^{h-1} + \gamma_i^p(t) & \text{if } L_{ij}/s^h < z_h - \gamma_i^p(t), h = h_t \\ (L_{ij} - l^{h-1})/s^h + z_h & \text{if } (L_{ij} - l^{h-1})/s^h < z_{h+1} - z_h, h \in \{h_t + 1, ..., h_\gamma\}, \end{cases} \tag{3}$$

where $l^{h_t-1} = s^{h_t}(z_{h_t} - t)$ and $l^h = l^{h-1} + s^h(z_{h+1} - z_h)$ if $h \in \{h_t + 1, ..., h_\gamma\}$, and $L_{ij} = \sum_{h=h_t}^{h_\gamma} l^h$. Note that the traversal time $\Gamma_p(t)$ of a path p can be induced from the arrival time at the destination node d. Therefore, it is given by:

$$\Gamma_p(t) = \gamma_d^p(t) - t. \tag{4}$$

3.1 Time-dependent GHG emission and fuel consumption functions

Our modeling for emissions and fuel consumption follows the same approach applied in some relevant works, including Bektaş and Laporte [3], Demir et al. [13], Franceschetti et al. [22, 23], Dabia et al. [8], Huang et al. [30] and Ehmke et al. [20]. According to these works GHG emissions are directly proportional to fuel consumption. We also use the CMEM with the parameters of Table 1 to estimate fuel consumption and GHG emissions. The CMEM is a microscopic model that allows the consideration of vehicle specific parameters, such as engine speed, traffic related parameters, and environment related factors [1, 2]. According to the CMEM the fuel use rate ($liter/s$) for a given time instant is a function encompassing travel speed, vehicle load and road gradient:

$$e_r = \frac{\zeta}{\varpi\psi}\left(kN_eV + \frac{1}{\varepsilon}\left(\frac{((w+q)(a + g\sin\theta + gC_r\cos\theta) + 0.5C_dA\rho s^2)s}{1000\eta_{tf}} + P_{acc}\right)\right). \tag{5}$$

All required parameters with their typical values are described in Table 1. P_{acc} is the engine power demand for vehicle accessories in hp. We consider the default value of P_{acc}, which is zero. Using $\alpha = a + g\sin\theta + gC_r\cos\theta$, $\beta = 0.5C_dA\rho$, $\varsigma = \frac{1}{(1000\varepsilon\eta_{tf})}$ and $\lambda = \frac{\zeta}{\varpi\psi}$, and based on the assumption associated with values of used parameters, expression (5) can be rewritten as:

$$e_r = \lambda(kN_eV + \varsigma\alpha(w+q)s + \varsigma\beta s^3). \tag{6}$$

Table 1: Parameters used in the CMEM

Notation	Description	Typical values
w	Curb-weight (kg)	15000
q	Carried load (kg)	0-10000
ζ	Fuel-to-air mass ratio	1
k	Engine friction factor ($kJ/rev/liter$)	0.25
N_e	Engine speed (rev/s)	60
V	Engine displacement ($liter$)	7
g	Gravitational constant (m/s^2)	9.81
ρ	Air density (k/m^3)	1.2041
C_d	Coefficient of aerodynamic drag	0.7
A	Frontal surface area (m^2)	5
C_r	Coefficient of rolling resistance	0.01
η_{tf}	Vehicle drive train efficiency	0.4
η	Efficiency parameter for diesel engines	0.9
c_f	Fuel and GHG emissions cost per liter ($CAD/$liter$)	1.05
c_d	Driver wage ($CAD/$s$)	0.0085
ϖ	Heating value of a typical diesel fuel (kJ/g)	44
ψ	Conversion factor (g/s to $liter/s$)	737
s^l	Lower speed limit (m/s)	8.333
s^u	Upper speed limit (m/s)	19.444
s	Average speed at a portion of segment (m/s)	
a	Acceleration (m/s^2)	0
θ	Roadway gradient (degree)	0

Using only average speeds of the whole day based on all observations across an arc or derived from a digital roadmap may not capture the impact of traffic on GHG emissions on a particular arc. In fact, CMEM computations taking into account only fixed speeds are often not accurate enough to reflect GHG emissions at peak hour traffic congestion considering fluctuating speed [17, 18, 43]. For example, if the travel speed on a path often drops far below the average speed, then the actual emissions may be much higher than if the trip occurs consistently at the average speed of the whole day. Thus, to optimize GHG emissions and the total cost in an urban area, one must explicitly consider the variability of the speed at different times of the day [18, 20].

To compute time-dependent arc costs we make use of individual speed observations. As in Fleischmann et al. [21] and Ehmke et al. [17] to account for time dependence, we group speed observations for an arc for each interval (06:00–06:15, 06:15–06:30, 06:30–06:45, etc.). Then, to address the challenge of the non-linearity of the emissions function we follow Dehne et al. [11] by using piecewise linear functions as an approximation of non-linear functions in deterministic contexts. More specifically, the costs change linearly according to speed variations when the boundary between consecutive intervals is crossed. Hence, we model time-dependent networks where emissions change linearly when the speed changes over an arc. Doing so, we compute the fuel consumption for traversing an arc starting at time t based on the FSM. For example, consider a vehicle which

enters an arc (i, j) of distance $L_{ij} = 1000$ meters at time $t_1 = 08{:}44$. For this simple scenario we assume that the speed on this arc is $s_{ij}^1 = 45$ km/h for the interval 08:30–08:45, then it decreases to $s_{ij}^2 = 35$ km/h at interval 08:30–08:45. Hence, we would use the speed value s_{ij}^1 associated with interval 08:30–08:45 for 60 seconds traversing 750 meters and the speed value s_{ij}^2 of interval 08:45–09:00 during 26 seconds traversing 250 meters to compute emissions and the total cost. In that case, the vehicle exits the arc at time $t_2 = 08{:}45{:}26$. So, the speed changes twice reflecting the variability in the speed of traffic. Therefore, for a given arc (i, j) along a path p starting at time t (time slot Z_{h_t}), the corresponding fuel consumption can be expressed based on a combination of equations (6) and (2):

$$\mathcal{F}_{ij}(t) = f_{ij}(t) + g_{ij}(t), \tag{7}$$

where

$$f_{ij}(t) = \sum_{h=h_t}^{h_\gamma} \left[\left(\frac{l_{ij}^h}{s_{ij}^h} \right) \lambda \varsigma \alpha (w+q) s_{ij}^h \right] = \lambda \varsigma \alpha (w+q) \sum_{h=h_t}^{h_\gamma} l_{ij}^h = \lambda \varsigma \alpha (w+q) L_{ij}, \tag{8}$$

and

$$g_{ij}(t) = \sum_{h=h_t}^{h_\gamma} \left[\left(\frac{l_{ij}^h}{s_{ij}^h} \right) \lambda (k N_e V + \varsigma \beta (s_{ij}^h)^3) \right] = \lambda k N_e V \tau_{ij}(t) + \lambda \varsigma \beta \sum_{h=h_t}^{h_\gamma} l_{ij}^h (s_{ij}^h)^2. \tag{9}$$

For a departure time t and a path p, the total amount of fuel consumed can be calculated as follows based on equation (7):

$$\mathcal{F}_p(t) = \sum_{(i,j) \in p} \mathcal{F}_{ij}(\gamma_i^p(t)). \tag{10}$$

3.2 Time-dependent travel cost function

Given a departure time t, the driver cost incurred from path p can be calculated as the cost of the traversal time according to (4):

$$\varphi(\Gamma_p(t)) = c_d \Gamma_p(t). \tag{11}$$

On the basis of (8) the cost of GHG emissions on a given arc (i, j) across a path p can be calculated as $c_f \mathcal{F}_{ij}(t)$. Thus, fuel consumption cost (in \$) of path p is given by:

$$\varphi(\mathcal{F}_p(t)) = c_f \mathcal{F}_p(t). \tag{12}$$

As the path's cost encompasses the traversal duration and fuel costs, the total cost of a path from

o to d starting at time t can be expressed by combining equations (11) and (12) as follows:

$$\varphi_p(t) = \varphi(\Gamma_p(t)) + \varphi(\mathcal{F}_p(t)) = \sum_{(i,j)\in p} c_{ij}(\gamma_i^p(t)), \tag{13}$$

where

$$c_{ij}(\gamma_i^p(t)) = c_d \tau_{ij}(\gamma_i^p(t)) + c_f f_{ij}(\gamma_i^p(t)). \tag{14}$$

Note that the cost $c_{ij}(\gamma_i^p(t))$ of traveling across arc (i,j) is expressed by combining equations (2) and (7).

An optimal path from source o to destination d given the starting time t is a path p_c^* with the least travel cost, denoted by $\varphi_{p_c^*}(t)$.

4 Time-dependent lower and upper bounds for the TDMCP-SO

When dealing with the TDMCP-SO, it is useful to compute the least cost lower and upper bounds to validate the accuracy of designed TDMCP-SO algorithms. Additionally, they can be embedded into time-dependent search heuristics as best-first search that associate to each node i a label equal to the known travel cost at arrival time $\gamma_i^p(t)$ at the current node i plus a lower bound on the cost to the destination.

This section presents time-dependent lower and upper bounds for the TDMCP-SO that can be computed by ignoring the network-wide traffic congestion. Let $\mathcal{P}_\varphi$ be the set of all feasible paths of the TDMCP-SO on $\mathcal{G}$. Given a path $p \in \mathcal{P}_\varphi$, let $\Gamma(p)$ be the traversal time of p assuming (1) holds. We also denote by $\underline{\Gamma}(p)$ the traversal time of p if the congestion ratios of all arcs are set to the lightest congestion factor $\sigma_h = \max_{(i,j)\in\mathcal{A}}(\sigma_{ijh})$ for each time slot Z_h. Let $\Delta = \min_{ijh}(\frac{s_{ij}^h}{\sigma_h u_{ij}})$ be the heaviest degradation of congestion ratio of any arc (i,j) over the time horizon $\mathcal{T}$. Finally, let $\underline{\underline{\Gamma}}(p)$ denote the duration of p if all s_{ij}^h are set to the speed limit u_{ij}. This is equivalent to assuming that all arc speeds become constant and the TDQPP (TDMCP-SO) reduces to the QPP (MCP-SO). Let p^*, $\underline{p}^*$ and $\underline{\underline{p}}^*$ be optimal solutions of the TDQPP under the assumptions previously defined, i.e., $p^* = \arg\min_{p\in\mathcal{P}_\varphi}\{\Gamma(p)\}$, $\underline{p}^* = \arg\min_{p\in\mathcal{P}_\varphi}\{\underline{\Gamma}(p)\}$, and $\underline{\underline{p}}^* = \arg\min_{p\in\mathcal{P}_\varphi}\{\underline{\underline{\Gamma}}(p)\}$.

Given a path $p \in \mathcal{P}_\varphi$, let $\varphi(p)$ be its traversal cost, starting at time t. We also denote $\underline{\underline{\varphi}}(p)$ the traversal cost of p if all s_{ij}^h are set to the speed limit u_{ij}. Let p_c^* and $\underline{\underline{p_c}}^*$ be optimal solutions of the TDMCP-SO and MCP-SO, respectively, thus, $p_c^* = \arg\min_{p\in\mathcal{P}_\varphi}\{\varphi(p)\}$, and $\underline{\underline{p_c}}^* = \arg\min_{p\in\mathcal{P}_\varphi}\{\underline{\underline{\varphi}}(p)\}$.

Adopting $\underline{\underline{p}}^*$ as a heuristic solution of the TDMCP-SO under the speed variation relationship (1) presents multiple advantages. Firstly, efficient algorithms designed for the QPP can be immediately applied to solve the TDMCP-SO. Secondly, if all arc speeds are set according to the maximum speed u_{ij}, then $\underline{\underline{p}}^*$ is a near-optimal solution for the TDMCP-SO. Indeed, in the following subsections we prove that $\varphi(\underline{\Gamma}(\underline{\underline{p}}^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*, \sigma_h)) \leq \varphi(p_c^*)$ is a lower bound on $\varphi(p_c^*)$ and that $\min\{\varphi(\underline{\underline{p_c}}^*), \varphi(p^*), \varphi(\underline{p}^*), \varphi(\underline{\underline{p}}^*)\} \leq \varphi(\underline{\underline{p}}^*)$ is its upper bound:

$$\varphi(\underline{\Gamma}(\underline{\underline{p}}^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*)) \leq \varphi(\Gamma(p^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*, \sigma_h)) \leq \varphi(p_c^*) \leq \min\{\varphi(\underline{\underline{p_c}}^*), \varphi(p^*), \varphi(\underline{p}^*), \varphi(\underline{\underline{p}}^*)\} \leq \varphi(\underline{\underline{p}}^*) \leq \tfrac{1}{\Delta}\varphi(\Gamma(p_c^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*)) \tag{15}$$

where $q_0 = 0$ is used to indicate empty load and $s^* = (\frac{kNV}{2\beta\varsigma})^{1/3}$ is the optimal speed which minimizes the fuel consumption cost for any arc, which results from $\frac{\partial \mathcal{F}_{ij}}{\partial s_{ij}}(s^*) = 0$.

4.1 A lower bound on the cost $\varphi(p_c^*)$

We now demonstrate that $\varphi(\underline{\Gamma}(\underline{\underline{p}}^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*)) \leq \varphi(\Gamma(p^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*, \sigma_h)) \leq \varphi(p_c^*)$ is a lower bound on $\varphi(p_c^*)$ and that if the vehicle travels with the same load q at speed s^* that minimizes fuel consumption across all arcs, then $\underline{\underline{p}}^*$ is optimal for TDMCP-SO, that is, $\underline{\underline{p}}^* = \underline{\underline{p_c}}^*$.

Theorem 1. *Path $\underline{\underline{p}}^*$ is an optimal solution for the TDMCP-SO when the vehicle travels with constant load, and the speed for all arcs $(i,j) \in \mathcal{A}$ is set to $s^* = (\frac{kNV}{2\beta\varsigma})^{1/3}$, minimizing fuel consumption.*

Proof. Given a solution path $p_c \in \mathcal{P}_\varphi$ it follows from (13) that:

$$\varphi(p_c, q, s^*) = \varphi(\Gamma(p_c, s^*)) + \varphi(\mathcal{F}(p_c, q, s^*)). \tag{16}$$

From (11) it results that the travel time can be expressed as (17) and fuel consumption and GHG emissions cost can be defined as (18), based on (8), (9) and (12):

$$\varphi(\Gamma(p_c, s^*)) = c_d \sum_{h=h_t}^{h_\gamma} \frac{l_{ij}^h}{s^*} = c_d \Gamma(p_c, s^*), \tag{17}$$

$$\varphi(\mathcal{F}(p_c, q, s^*)) = c_f(\lambda\varsigma\alpha w s^* + \lambda(kN_eV + \varsigma\beta(s^*)^3)) \sum_{h=h_t}^{h_\gamma} \frac{l_{ij}^h}{s^*} = c_f e_r(q, s^*)\Gamma(p_c, s^*), \tag{18}$$

where $e_r(q, s^*)$ is constant across all arcs of path p_c and represents the minimum fuel consumption rate. Combining (17) and (18) results in:

$$\varphi(p_c, q, s^*) = [c_d + c_f e_r(q, s^*)]\, \Gamma(p_c, s^*). \tag{19}$$

As the vehicle travels with load q at speed s^* across all arcs, the first part of the equation (19) $c_d + c_f e_r(q, s^*)$ is constant. Hence, the overall cost is minimum if the total travel time $\Gamma(p_c, s^*)$ of the path p_c is minimum. The minimum travel time is given by an optimal solution for the TDQPP, i.e., $\underline{\underline{p}}^*$. Hence, an implication of (19) is that the optimal path $p_c^* = \underline{\underline{p}}^*$, which completes the proof of Theorem 1.

Theorem 2. *Given two optimal paths $\underline{\underline{p}}^*$ and p_c^* (with respect to $\varphi(\underline{\underline{p}}^*)$ and $\varphi(p_c^*)$, respectively) for the TDQPP and the TDMCP-SO, respectively, the following relationship is satisfied:*

$$\varphi(\underline{\Gamma}(\underline{\underline{p}}^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*)) \leq \varphi(\Gamma(p^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*, \sigma_h)) \leq \varphi(p_c^*). \tag{20}$$

Proof. By observing that when the congestion ratios of all arcs are set to their lightest values $\sigma_h = \max_{(i,j)\in\mathcal{A}} (\sigma_{ijh})$ for each time interval Z_h it follows that the traversal time of a given path $p = (o = v_0, v_2, ..., v_k = d)$ starting a time $t = t_0$ is

$$\underline{\underline{\Gamma}}(p) = \sum_{l=1}^{k} \frac{L_{v_{l-1}v_l}}{u_{v_{l-1}v_l}} = \sum_{l=1}^{k} \int_{t}^{t+\tau_{v_{l-1}v_l}(t)} \sigma(\gamma) d\gamma = \int_{t_0}^{\underline{\Gamma}(p)} \sigma(\gamma) d\gamma \tag{21}$$

where $\sigma(\gamma) = \sigma_h$. Furthermore, if we consider another path $p' \in \mathcal{P}_\varphi$, from (21) it follows that:

$$\underline{\underline{\Gamma}}(p') \leq \underline{\underline{\Gamma}}(p) \Leftrightarrow \underline{\Gamma}(p') \leq \underline{\Gamma}(p), \tag{22}$$

which implies that:

$$\underline{\Gamma}(\underline{\underline{p}}^*) = \underline{\Gamma}(\underline{p}^*). \tag{23}$$

As $\underline{p}^* = \arg\min_{p\in\mathcal{P}_\varphi} \{\underline{\Gamma}(p)\}$, $\underline{\Gamma}(p_c^*) \leq \Gamma(p_c^*)$, and $\Gamma(p^*) \leq \Gamma(p_c^*)$, from (23) it results that:

$$\underline{\Gamma}(\underline{\underline{p}}^*) \leq \Gamma(p^*) \leq \Gamma(p_c^*). \tag{24}$$

Hence,

$$\varphi(\underline{\Gamma}(\underline{\underline{p}}^*)) \leq \varphi(\Gamma(p^*)) \leq \varphi(\Gamma(p_c^*)). \tag{25}$$

If we consider the case of fixed speed s^* which minimizes the fuel consumption we may assert that:

$$\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*) \leq \mathcal{F}(\underline{\underline{p}}^*, q_0, s^*, \sigma_h) \leq \mathcal{F}(p_c^*). \tag{26}$$

Then the following relationship also holds:

$$\varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*)) \leq \varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*, \sigma_h)) \leq \varphi(\mathcal{F}(p_c^*)). \tag{27}$$

Combining (25) and (27) yields:

$$\varphi(\underline{\Gamma}(\underline{\underline{p}}^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*)) \leq \varphi(\Gamma(p^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*, \sigma_h)) \leq \varphi(p_c^*), \tag{28}$$

which completes the proof of Theorem 2.

4.2 A worst case analysis

In this subsection, we provide a worst case analysis on the cost $\varphi(\underline{\underline{p}}^*)$.

Theorem 3. *The value $\varphi(\underline{\underline{p_c}}^*)$ is an upper bound not greater than $\frac{1}{\Delta}\varphi(\Gamma(p^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*))$.*

Proof. As p^*, $\underline{p}^*$, and $\underline{\underline{p}}^*$ are optimal solutions for the TDQPP, they are also feasible solutions for the TDMCP-SO. Additionally, $\underline{\underline{p_c}}^*$ is a feasible solution for the TDMCP-SO, then:

$$\varphi(p_c^*) \leq \varphi(p^*) \tag{29}$$

$$\varphi(p_c^*) \leq \varphi(\underline{p}^*) \tag{30}$$

$$\varphi(p_c^*) \leq \varphi(\underline{\underline{p}}^*) \tag{31}$$

$$\varphi(p_c^*) \leq \varphi(\underline{\underline{p_c}}^*). \tag{32}$$

By combining (29), (30), (31) and (32) we obtain the proof of the first part of the upper bound inequality:

$$\varphi(p_c^*) \leq \min\{\varphi(\underline{\underline{p_c}}^*), \varphi(p^*), \varphi(\underline{p}^*), \varphi(\underline{\underline{p}}^*)\} \leq \varphi(\underline{\underline{p}}^*) \tag{33}$$

Furthermore, if the congestion ratio σ_{ijh} takes a value in the interval $[\Delta\sigma_h, \sigma_h]$, then it results that for a given path p:

$$\underline{\Gamma}(p) \leq \Gamma(p) \leq \frac{1}{\Delta}\underline{\Gamma}(p). \tag{34}$$

Combining (33) and (34) yields:

$$\Gamma(\underline{\underline{p}}^*) \leq \frac{1}{\Delta}\Gamma(p^*) \leq \frac{1}{\Delta}\Gamma(p_c^*), \tag{35}$$

which implies that:

$$\varphi(\Gamma(\underline{\underline{p}}^*)) \leq \frac{1}{\Delta}\varphi(\Gamma(p^*)) \leq \frac{1}{\Delta}\varphi(\Gamma(p_c^*)). \tag{36}$$

Since $\varphi(\underline{\underline{p}}*) = \varphi(\Gamma(\underline{\underline{p}}^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*))$, it follows that:

$$\varphi(\Gamma(\underline{\underline{p}}^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*)) \leq \frac{1}{\Delta}\varphi(\Gamma(p_c^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*)). \tag{37}$$

An implication of (37) is that $\varphi(\underline{\underline{p}}*)) \leq \frac{1}{\Delta}\varphi(\Gamma(p_c^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*))$, which completes the proof of Theorem 3.

5 Design of efficient TDMCP-SO algorithms

In this section, we propose several heuristic algorithms that will be applied to efficiently solve the TDMCP-SO in polynomial time based either on the FSM, LTM or LTM-STTF. First, we present the time-dependent Dijkstra algorithm. Second, we present the algorithms used to compute arrival times, travel times, emissions and travel costs. Lastly, we describe speed-up methods for the fast computation of lower and upper bounds on path traversal costs.

5.1 Time-dependent Dijkstra algorithms

To solve the TDMCP-SO we propose new solution methods based on three adaptations of Dijkstra's label-setting algorithm presented in Dean [10]. LTM, LTM-STTF and FSM models for the computation of time-dependent arc arrival time, fuel consumption and travel costs are integrated at every iteration of the main label-setting algorithm when choosing the next connecting node. We call these modified versions the time-dependent Dijkstra's (TD-Dijkstra) FSM, LTM and LTM-STTF algorithms.

We note that LTM does not ensure FIFO consistency, but does not allow for cycles to exist [18]. As arc travel times are specified upon entrance at the head node of the arc and are assumed to be fixed for that particular vehicle until it leaves the terminal node [34]. LTM is also referred to as the frozen link model [36]. That is, if a subsequent interval is crossed while traversing an arc, additional steps are needed to ensure that the transition between the intervals is seamless to guarantee the FIFO property [21]. Doing so, Ehmke et al. [17] and Ehmke et al. [20] linearized the different speed levels between neighboring intervals in the transition area using a STTF as proposed in Fleischmann et al. [21].

Clearly, both LTM and LTM-STTF map the approximation of the travel time to one of the time intervals considering the entering time to the head of the arc. However, our approach considers the set of time periods crossed when traversing the arc in the computation of emissions and travel cost. In this case the emissions and travel cost across each arc are computed according to the set of speeds obtained at the time of traversing the arc, which ensures FIFO consistency. In fact, FSM handles every speed level changes when traveling across an arc considering a small or long distance. Hence, with FSM no preprocessing steps are needed to guarantee FIFO property, which is more realistic than LTM that requires additional steps to ensure FIFO consistency using smoothing methods.

Let o denote the origin node and $predecessor(i)$ be the predecessor of node i. Therefore, the TD-Dijkstra-* label-setting algorithms are designed as in Algorithm 1. The TD-Dijkstra-FSM and TD-Dijkstra-LTM algorithms work by examining all temporarily labeled nodes in the network starting with the source node o. At the beginning, the priority queue $\mathcal{N}$ contains all nodes and their status are initialized to unlabeled except the source o. Hence, c_o is set to 0 and for each unlabeled node i the cost c_i is set to ∞. At each iteration of node expansion, the algorithm selects a labeled but not examined node i with the least labeled time-dependent cost from the set of temporarily labeled nodes $\mathcal{E}^+(i)$, updates its cost label, and puts the node into a set of examined

and permanently labeled nodes $\mathcal{E}$, and each arc leaving from it is evaluated. If the labeled cost of node i plus the cost of arc (i, j) is smaller than the labeled cost of node j, then the cost from the source node to node j is updated with a value equal to the sum of the labeled cost of node i plus the cost of arc (i, j). Then, the algorithm continues the node examination process and takes the next node to be examined. The algorithm terminates when the destination node d is reached or when the priority queue becomes empty. In the node-examination process, a tree connecting all examined nodes is created, and the permanently labeled time-dependent travel cost associated with each examined node represents the least cost path from the origin node to that one.

The proposed label-setting algorithm maintains one label for each node to include travel cost value for the expected minimum-cost path starting from the source to the node associated with the label. By employing one label for each node our solution method reduces the multiplication of labels and computation time. The algorithm operates on all labels that are updated in step 15 of Algorithm 1 to define a pointer back to the node associated with it to allow the tracking of the minimum cost path.

As shown in steps 14 and 15 of Algorithm 1 the label-setting algorithm and the travel cost computation function have been combined inside the same procedure to reduce computational complexity. Thus, the cost calculation function is executed at the next nodes connecting the current one. Let $Travel\_Cost\_*()$ be the method that computes the travel cost at node j when starting from node i at time $\gamma_i^p(t)$, where the symbol $*$ represents the appropriate model for time-dependent networks (FSM, LTM or LTM-STTF).

The original Dijkstra's algorithm has a computational complexity of $O(m \log(n))$. However, the TD-Dijkstra-LTM has a computational complexity of $O(m \log(n) + n)$, where n and m are the number of nodes and arcs in the time-dependent network, respectively. For every arc and departure time at node i the arc travel cost is computed in $O(1)$. As each node label remembers the index of the time period, we reduce the scanning time from $O(m)$ to $O(n)$. With the TD-Dijkstra-LTM-STTF the travel times are derived from the original data using STTF to overcome the passing that may occur when the travel time decreases at some time intervals. Hence, the computational complexity becomes $O(m \log(n) + nC)$, where C is the number of steps used by STTF. Similarly, the TD-Dijkstra-FSM algorithm solves the TDMCP-SO with $O(m \log(n) + n\mathcal{K})$ time complexity, where $\mathcal{K}$ denotes the maximum number of time periods scanned by the function Travel_Cost_FSM.

Algorithm 1 Determination of a near-optimal least cost path by adapting Dijkstra label-setting algorithm (TD-Dijkstra-*)

function TD_DIJKSTRA($o, d, \mathcal{G}$)
    $\mathcal{E} \leftarrow \varnothing$
    $\mathcal{N} \leftarrow \mathcal{V}$
    $c_i \leftarrow \infty$, $\forall i \in \mathcal{V}$
    $c_o \leftarrow 0$ and $predecessor(o) \leftarrow o$
    while $|\mathcal{E}| < n$ **do**
        let $i \in \mathcal{N}$ be a node for which $c_i \leftarrow \min\{c_j : j \in \mathcal{N}\}$
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{i\}$
        $\mathcal{N} \leftarrow \mathcal{N} \backslash \{i\}$
        if $i = d$ **then**
            Stop
        end if
        for each $(i, j) \in \mathcal{E}^+(i)$ **do**
            if $c_j > Travel\_Cost\_*(\gamma_i^p(t), (i, j), \mathcal{Z}, \mathcal{S})$ **then**
                $c_j \leftarrow c_i + Travel\_Cost\_*(\gamma_i^p(t), (i, j), \mathcal{Z}, \mathcal{S}))$
                $predecessor(j) \leftarrow i$
            end if
        end for
    end while
end function

5.2 Time-dependent arrival time and travel time computation

In the case of the FSM, during each time period Z_h the flow speed on each arc (i, j) is assumed to be constant. Given the set of speeds and a starting time $\gamma_i^p(t)$ at node i, both arrival and travel times across arc (i, j) can be computed through Algorithm 2.

Algorithm 2 Computing the travel time $\tau_{ij}(\gamma_i^p(t))$ across a given arc (i, j) based on the FSM

function TRAVEL_TIME_FSM($\gamma_i^p(t)$, $(i, j), \mathcal{Z}, \mathcal{S}$)
  $h|\gamma_i^p(t) \in Z_h = [z_h, z_{h+1}[$
  $k \leftarrow h$
  $d \leftarrow L_{ij} - \left[s_{ij}^k(z_{k+1} - \gamma_i^p(t))\right]$
  while $d > 0$ **do**
    $k \leftarrow k + 1$
    $d \leftarrow d - \left[s_{ij}^k(z_{k+1} - z_k)\right]$
  end while
  $\gamma_j^p(t) \leftarrow z_{k+1} + d/s_{ij}^k$
  $\tau_{ij}(\gamma_i^p(t)) \leftarrow \gamma_j^p(t) - \gamma_i^p(t)$
  return $\tau_{ij}(\gamma_i^p(t))$
end function

In the case of the LTM the travel time of arc (i, j) is specified when departing from node i at a given time period Z_h and is assumed to be constant until exiting at the node j. The calculation of arrival and travel times across arc (i, j) are summarized in Algorithm 3.

Algorithm 3 Computing the travel time $\tau_{ij}(\gamma_i^p(t))$ across a given arc (i, j) based on the LTM

function TRAVEL_TIME_LTM($\gamma_i^p(t)$, $(i, j), \mathcal{Z}, \mathcal{S}$)
  $h|\gamma_i^p(t) \in Z_h = [z_h, z_{h+1}[$
  $\tau_{ij}(\gamma_i^p(t)) \leftarrow L_{ij}/s_{ij}^h$
  return $\tau_{ij}(\gamma_i^p(t))$
end function

5.3 Time-dependent fuel consumption and travel cost computation

Given a starting time $\gamma_i^p(t)$ at node i, the fuel consumption, GHG emissions and travel costs across arc (i, j) are computed using the FSM and the LTM models based on Algorithms 4 and 5,

respectively.

In Algorithm 4, we identify all speed changes according to the time periods crossed when traversing arc (i, j) and consider the associated portions of distance covered. Hence, at every iteration the time-dependent travel cost and energy consumption are induced, including the amount of GHG emissions and fuel consumption computed using CMEM. The algorithm stops when node j is reached.

Algorithm 4 Computing the travel cost $c_{ij}(\gamma_i^p(t))$ across a given arc (i, j) based on the FSM

function TRAVEL_COST_FSM($\gamma_i^p(t)$, $(i, j), \mathcal{Z}, \mathcal{S}$)
  $h|\gamma_i^p(t) \in Z_h = [z_h, z_{h+1}[$
  $k \leftarrow h$
  $l \leftarrow s_{ij}^k(z_{k+1} - \gamma_i^p(t))$
  $d \leftarrow L_{ij} - l$
  $g \leftarrow \lambda k N_e V \left(\frac{l}{s_{ij}^k}\right) + l\lambda\varsigma\beta(s_{ij}^k)^2$
  while $d > 0$ **do**
    $k \leftarrow k + 1$
    $l \leftarrow s_{ij}^k(z_{k+1} - z_k)$
    $g \leftarrow g + \lambda k N_e V \left(\frac{l}{s_{ij}^k}\right) + l\lambda\varsigma\beta(s_{ij}^k)^2$
    $d \leftarrow d - \left[s_{ij}^k(z_{k+1} - z_k)\right]$
  end while
  $\gamma_j^p(t) \leftarrow z_{k+1} + d/s_{ij}^k$
  if $k > h$ **then**
    $l \leftarrow s_{ij}^k(\gamma_j^p(t) - z_k)$
    $g \leftarrow g + \lambda k N_e V \left(\frac{l}{s_{ij}^k}\right) + l\lambda\varsigma\beta(s_{ij}^k)^2$
  else
    $g \leftarrow \lambda k N_e V \left(\frac{L_{ij}}{s_{ij}^h}\right) + L_{ij}\lambda\varsigma\beta(s_{ij}^h)^2$
  end if
  $\tau_{ij}(\gamma_i^p(t)) \leftarrow \gamma_j^p(t) - \gamma_i^p(t)$
  $\mathcal{F}_{ij}(\gamma_i^p(t)) \leftarrow \lambda\varsigma\alpha(w + q)L_{ij} + g$
  $c_{ij}(\gamma_i^p(t)) \leftarrow c_d\tau_{ij}(\gamma_i^p(t)) + c_f\mathcal{F}_{ij}(\gamma_i^p(t)$
  return $c_{ij}(\gamma_i^p(t))$
end function

Algorithm 5 Computing the travel cost $c_{ij}(\gamma_i^p(t))$ across a given arc (i,j) based on the LTM

function TRAVEL_COST_LTM($\gamma_i^p(t)$, $(i,j), \mathcal{Z}, \mathcal{S}$)

$h|\gamma_i^p(t) \in Z_h = [z_h, z_{h+1}[$

$\tau_{ij}(\gamma_i^p(t)) \leftarrow L_{ij}/s_{ij}^h$

$\gamma_j^p(t) = \gamma_i^p(t) + \tau_{ij}(\gamma_i^p(t))$

$\mathcal{F}_{ij}(\gamma_i^p(t)) \leftarrow \lambda(kN_eV + \varsigma\alpha(w+q)s_{ij}^h + \varsigma\beta(s_{ij}^h)^3)\frac{L_{ij}}{s_{ij}^h}$

$c_{ij}(\gamma_i^p(t)) \leftarrow c_d\tau_{ij}(\gamma_i^p(t)) + c_f\mathcal{F}_{ij}(\gamma_i^p(t))$

return $c_{ij}(\gamma_i^p(t))$

end function

5.4 Dijkstra with speed limits and fast computation of time-dependent least cost upper and lower bounds

We now propose an effective speed-up technique to ensure the fast computation of time-dependent lower and upper bounds for the TDMCP-SO. First, we run the classical Dijkstra to solve the TDMCP-SO to optimality where the network-wide traffic congestion is ignored and travel speeds become constant. We call this version the Dijkstra algorithm with speed limits (Dijkstra-SL). In this case, all arc travel speeds s_{ij}^h are set according to the speed limit u_{ij}. Second, we compute the lower bound by applying Algorithms 6 and 7 in order to obtain travel costs across each arc of the MCP-SO optimal solution considering the appropriate set of speeds. As shown in step 3 of Algorithm 6 driver costs are computed using the method described in Algorithm 2, considering the heaviest congestion ratio σ_h for all arcs over time interval Z_h. Algorithm 7 computes fuel cost considering optimal speeds (see Section 4). Next, we compute upper bounds by evaluating the travel cost across each arc using Algorithm 4 considering time-varying speeds. Compared to Ehmke et al. [17] we considered the set of lightest congestion factors for different time periods and optimal speed to compute the lower bounds.

Algorithm 6 Computing the driver cost across a given arc (i,j) according to the heaviest congestion ratios

function DRIVER_COST_HEAVIEST($\gamma_i^p(t)$, $(i,j), \mathcal{Z}, \sigma_h$)

$h|\gamma_i^p(t) \in Z_h = [z_h, z_{h+1}[$

$\tau_{ij}(\gamma_i^p(t)) \leftarrow Travel\_Time\_FSM(\gamma_i^p(t), (i,j), \mathcal{Z}, \sigma_h)$

return $c_d\tau_{ij}(\gamma_i^p(t))$

end function

Algorithm 7 Computing the fuel cost across a given arc (i, j) based on optimal speeds

function FUEL_COST_OPTIMAL_SPEED($\gamma_i^p(t)$, $(i, j), \mathcal{Z}, s^*$)

  $h|\gamma_i^p(t) \in Z_h = [z_h, z_{h+1}[$

  $\tau_{ij}(\gamma_i^p(t)) \leftarrow L_{ij}/s^*$

  $\mathcal{F}_{ij}(\gamma_i^p(t)) \leftarrow \lambda(kN_eV + \varsigma\alpha(w+q)s^* + \varsigma\beta(s^*)^3)\tau_{ij}(\gamma_i^p(t))$

  return $c_f\mathcal{F}_{ij}(\gamma_i^p(t))$

end function

6 Computational experiments

In this section, the experimental design and methodology for generating networks with their arc information are provided. Then, detailed computational results of our TDMCP-SO algorithms are presented and analyzed.

6.1 Benchmarks set

Our experiments are conducted on a real large road network generated from the geographical information of Québec City. The obtained network contains 50,367 arcs and 17,431 nodes, and is composed by a set of physical nodes, and a set of arcs of different types, such as arterial streets, ramps and highway segments. Figure 2 shows a portion of the geographical area. We have considered 60 time periods of 15 minutes from 6h00 to 21h00, which covers a typical workday. For each arc and each time period, the time-dependent flow speeds are computed based on a large set of real-world data including more than 24 million of GPS observations provided by the city administration and logistic partners [4]. More specifically, to obtain historical congestion ratios for each road segment in this network, the time-varying speed dataset was analyzed using geomatics and geospacial manipulations by geomatic specialists to match each speed observation with the GPS coordinates connecting each road links. Note that the arcs of the considered road network are relatively short with an average of 158.08 meters, as our network is based on a dense urban city.

As shown in Table 2 we have designed 400 test instances divided into four sets:

1. large networks considering a fixed departure time,
2. medium networks considering a fixed departure time,
3. small networks considering a fixed departure time,

4. and large networks with different departure times.

For each instance we generate a pair of source and destination which corresponds to real historical shipment data provided by one of our logistic partners. Further, each instance is solved with different carried loads: empty (15 tons), less-than-truck load (LTL – 17.5, 20, and 22.5 tons) and full truck load (TL – 25 tons).

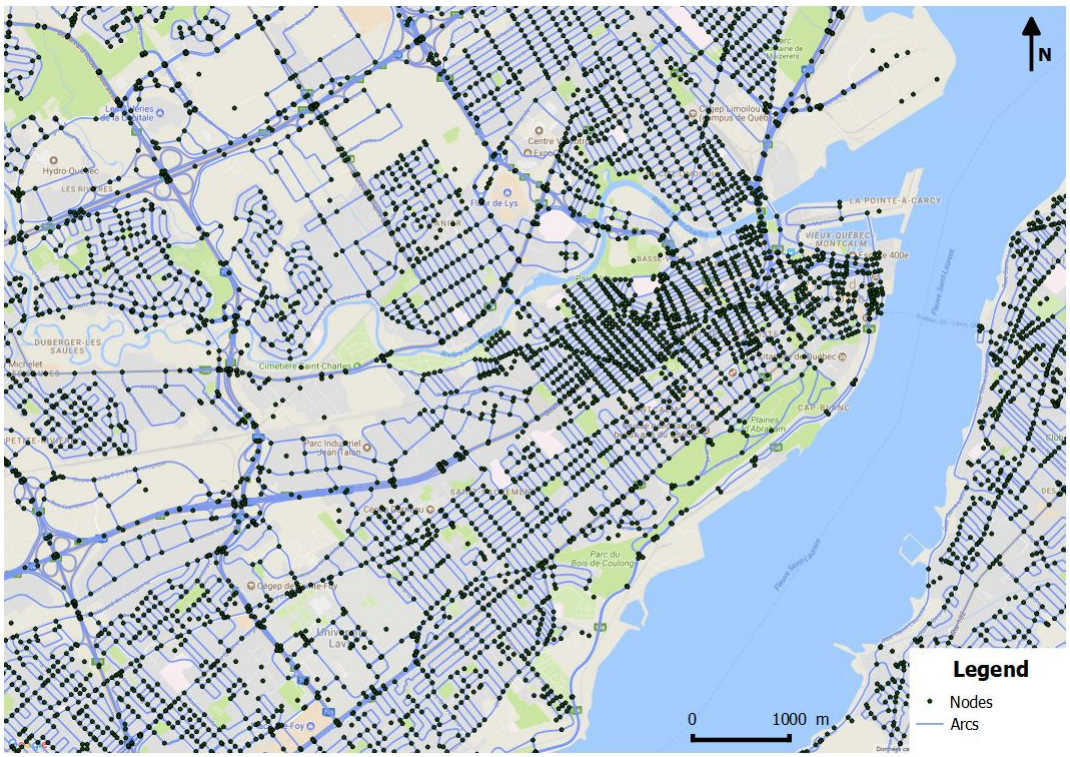

Figure 2: Portion of the geographical area

6.2 Experimental design

Table 3 summarizes our experimental design. All the instances were solved using three different optimization objectives, namely travel time, fuel consumption, and travel cost. Observe that minimizing fuel consumption is equivalent to minimizing GHG emissions as one liter of diesel generates 0.00279 $t\ CO_2\ e$ [35]. For each set of instances and objective functions, we apply the developed algorithms by adjusting their objective function accordingly, namely classical Dijkstra with speed limits (Dijkstra-SL), TD-Dijkstra-LTM, TD-Dijkstra-LTM-STTF, TD-Dijkstra-FSM, LB and UB. Then, the exact value of each solution is recalculated with Algorithms 2 and 4 according to FSM and CMEM to reflect the key elements of real road network considering time-varying speeds.

Table 2: Test instances

Instances	Networks	Number of nodes	Number of arcs	Speed observations	Departure time	Carried load (ton)
L1-L20	Large	17431	50367	613485		
M1-M20	Medium	3859	5388	266280	08h15	
S1-S20	Small	1612	2810	78709		
D1					07h30	
D2					08h00	
D3					08h30	
D4					09h00	
D5					09h30	
D6					10h00	
D7					10h30	15.0
D8					11h00	17.5
D9					11h30	20.0
D10					12h00	22.5
D11	Large	17431	50367	613485	12h30	25.0
D12					13h00	
D13					13h30	
D14					14h00	
D15					14h30	
D16					15h00	
D17					15h30	
D18					16h00	
D19					16h30	
D20					17h00	

All algorithms are implemented in C++ 17 using Jetbrains CLion C++ 2018 release 2.5 with cmake C++ compiler and were run on a ThinkCenter professional workstation with 32-gigabyte RAM and Intel core i7 vPro, running Ubuntu Linux 16.05 LTS x86 operating system.

Table 3: Overview of experimental design

Optimization criteria	Related routing problems	Algorithms	Solution evaluation criteria
Travel time Fuel (GHG emission) Cost	TDQPP TDLEPP TDMCP-SO	Dijkstra-SL TD-Dijkstra-LTM TD-Dijkstra-LTM-STTF TD-Dijkstra-FSM LB/UB	Distance (m) Travel time (s) Fuel consumption ($liter$) Cost ($)

6.3 Computational results and analysis

In this section we assess the effectiveness and robustness of the proposed algorithms on an extensive set of benchmarks showing the differences between LTM, LTM-STTF and FSM. Table 4 shows the average results of the proposed algorithms for each of the three optimization criteria over the four sets of instances. For each combination we present the average distance in meters (Dist), the travel time in seconds (TT), the fuel consumption in liters (Fuel), the total cost in dollars (Cost), and the required computing time in seconds (Sec). For these results we assume the case of full truck load (25 t).

As shown in Table 4, the results indicate that the proposed algorithms run quickly even for very large size instances. Such fast solution is critical for providing real-time routing to drivers. For the cost optimization objective, the average computation time of TD-Dijkstra-LTM, TD-Dijkstra-LTM-STTF and TD-Dijkstra-FSM is 0.43, 054 and 0.33 seconds, respectively. Hence, regarding computational efficiency the TD-Dijkstra-FSM outperforms TD-Dijkstra-LTM and TD-Dijkstra-LTM-STTF by 30.30% and 63.64%, respectively. Clearly, using a FIFO-consistent method reduces run times as it requires fewer steps to evaluate travel times and travel costs. Further, we see from Table 5 that our TD-Dijkstra-FSM yields a global saving on the number of node scanning of 0.017% (12110 versus 12112) and 0.21% (12110 versus 12135), respectively, between TD-Dijkstra-FSM, TD-Dijkstra-LTM and TD-Dijkstra-LTM-STTF. Note that the computation time of the

proposed algorithms is less than one second for all instances. To further assess the performance and the scalability of the algorithms various experiments have been performed with road networks provided by the 9th DIMACS challenge for the classical SPP. The average computation time for each core instance and over 1000 random node pairs were collected. As an example, the full USA road network instance includes 23.947 million nodes and 58.333 million arcs. The average runtime of TD-Dijkstra-FSM algorithm is 4.8 seconds for 20 DIMACS instances with the full USA road network.

Regarding the quality of the obtained solutions, we first observe that the TD-Dijkstra-FSM generates the best solutions for each optimization criterion. More specifically, when we minimize the travel time, TD-Dijkstra-FSM yields an optimal solution for each instance under FIFO networks using time-varying speeds. For the Travel Time optimization criterion, Table 4 shows that over our 80 instances TD-Dijkstra-FSM produces an average travel time of 1,338.59 seconds, which is 6.67% lower than Dijkstra-SL (1,434.25 seconds), 0.91% lower than TD-Dijkstra-LTM (1,350.79 seconds), and 0.73% lower than TD-Dijkstra-LTM-STTF (1,348.40 seconds). This exposes the error associated with using the speed limit, LTM or LTM-STTF instead of using calculations with the FSM. On the one hand, LTM-STTF integrates a smoothing method into LTM to ensure FIFO consistency. On the other hand, with the FSM the speed on each arc depends on the time interval, and the arrival times are consistent with the FIFO property. The fuel consumption reported by the TD-Dijkstra-FSM (under the travel time objective) is 9.90 $liters$ of fuel for a distance of 20.82 km which corresponds to 47.54 $liters$ per 100 km. This value is remarkably close to the annual average consumption of 46.9 reported by Transports Canada [42] for heavy duty vehicles with more than 4500 kg.

When looking at each optimization criterion, the TD-Dijkstra-FSM algorithm minimizes the travel time and the fuel consumption, as expected yielding the best results. For example, when Fuel Consumption is minimized the TD-Dijkstra-FSM reduces the travel time (1440.63 seconds), on average, by 16.21% (1674.16 seconds), 1.25% (1458.58 seconds) and 0.56% (1448.79 seconds) compared to Dijkstra-SL, TD-Dijkstra-LTM, and TD-Dijkstra-LTM-STTF, respectively, decreasing the fuel consumption from 9.82, 9.37, and 9.36 to 9.34 liters. These results clearly show that our TD-Dijkstra-FSM algorithm successfully manages avoiding traffic congestion to find better fuel consumption minimizing paths. Let us look at Figure 3 showing an example of the paths obtained by the time-dependent algorithms for the instance L20 under the cost optimization objective. It

Table 4: Algorithm performances under different optimization criteria

Optimization criteria	Instances	Dijkstra-SL					TD-Dijkstra-LTM					TD-Dijkstra-LTM-STTF					TD-Dijkstra-FSM				
		Dist	TT	Fuel	Cost	Sec	Dist	TT	Fuel	Cost	Sec	Dist	TT	Fuel	Cost	Sec	Dist	TT	Fuel	Cost	Sec
Travel Time	***L1-L20***	24310.27	1770.60	12.08	28.95	0.19	24845.29	1687.35	11.98	28.12	0.44	24732.45	1671.35	11.90	27.89	0.49	24901.75	**1658.10**	11.90	27.78	0.39
	M1-M20	10397.11	990.20	5.79	15.07	0.11	10886.42	875.9	5.57	13.79	0.26	10980.47	877.25	5.60	13.90	0.28	10946.48	**867.05**	5.57	13.78	0.24
	S1-S20	6216.59	664.05	3.68	9.87	0.06	6910.06	569.85	3.57	8.94	0.15	7328.93	579.50	3.73	9.22	0.15	6846.87	**565.45**	3.53	8.87	0.13
	D1-D20	42174.18	2312.15	19.28	41.83	0.24	39288.77	2270.05	18.13	40.36	0.56	40559.14	2265.50	18.56	40.60	0.62	40593.86	**2263.75**	18.59	40.62	0.50
	Average	20774.54	1434.25	10.21	23.93	0.15	20482.64	1350.79	9.81	22.80	0.35	20900.25	1348.40	9.95	22.90	0.38	20822.24	**1338.59**	9.90	22.76	0.32
Fuel Consumption	***L1-L20***	22321.74	2016.95	12.01	30.95	0.19	23072.33	1747.90	11.52	28.10	0.50	23081.68	1745.7	11.51	28.08	0.61	23094.44	1734.90	**11.49**	27.95	0.40
	M1-M20	9416.36	1110.45	5.80	16.11	0.11	9992.72	919.20	5.38	13.99	0.29	10029.95	909.75	5.36	13.89	0.34	9994.84	910.95	**5.35**	13.90	0.23
	S1-S20	5701.79	715.25	3.64	10.26	0.06	6022.40	621.05	3.42	9.22	0.15	6040.07	612.45	3.40	9.12	0.18	6034.97	605.80	**3.38**	9.03	0.12
	D1-D20	34468.35	2854.00	17.84	44.77	0.26	35092.44	2546.15	17.17	41.39	0.66	35289.49	2527.25	17.18	41.23	0.79	35265.19	2510.85	**17.14**	41.05	0.52
	Average	17977.06	1674.16	9.82	25.53	0.16	18544.97	1458.58	9.37	23.17	0.40	18610.3	1448.79	9.36	23.08	0.48	18597.36	1440.63	**9.34**	22.98	0.32
Cost	***L1-L20***	23208.80	1834.30	11.89	29.26	0.19	23970.88	1681.00	11.67	27.70	0.54	23820.56	1684.1	11.63	27.69	0.69	23935.03	1672.30	11.64	**27.59**	0.42
	M1-M20	9783.68	1052.75	5.79	15.60	0.11	10664.25	873.15	5.49	13.74	0.31	10649.6	875.95	5.49	13.76	0.39	10669.63	870.20	5.48	**13.70**	0.24
	S1-S20	5784.21	702.65	3.64	10.15	0.06	6651.81	572.50	3.50	8.89	0.17	6625.70	574.00	3.49	8.90	0.21	6668.82	568.75	3.49	**8.85**	0.13
	D1-D20	35977.58	2483.40	17.57	41.31	0.25	37154.89	2305.50	17.47	39.68	0.70	37018.2	2311.05	17.42	39.68	0.86	36925.96	2313.10	17.40	**39.67**	0.55
	Average	18688.57	1518.28	9.72	24.08	0.15	19610.46	1358.04	9.53	22.50	0.43	19528.51	1361.28	9.51	22.51	0.54	19549.86	1356.09	9.50	**22.45**	0.33

is easy to see that path obtained by each algorithm may be different based on the way that each method compute cost.

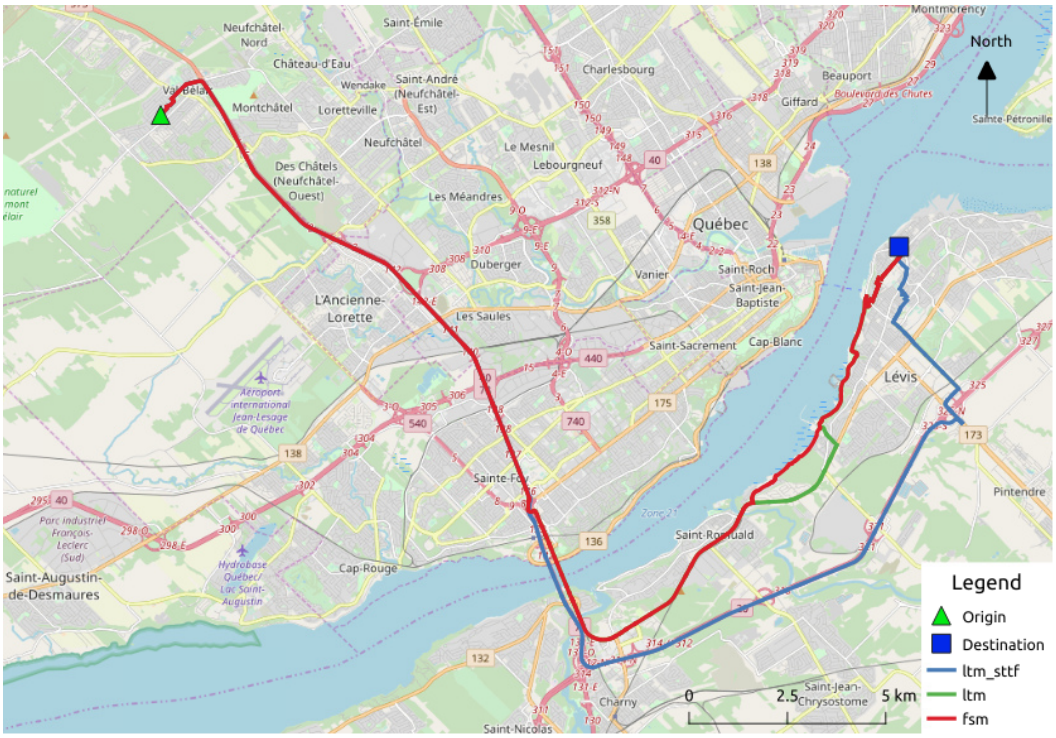

Figure 3: Paths obtained with TD-Dijkstra-LTM, TD-Dijkstra-LTM-STTF, and TD-Dijkstra-FSM for the L20 instance under the cost optimization objective

The third part of Table 4 considers the cost-minimizing objective. Again, the TD-Dijkstra-FSM produces the best results with an average path cost of 22.45$ compared to 24.08$, 22.50$ and 22.51$ with Dijkstra-SL, TD-Dijkstra-LTM and TD-Dijkstra-LTM-STTF. Minimizing the cost implies a compromise between the travel time (cost) of the drivers and the fuel cost. Compared to Dijkstra-SL, TD-Dijkstra-LTM and TD-Dijkstra-LTM-STTF solutions, the one obtained with TD-Dijkstra-FSM under the cost minimization criterion has better travel time than when minimizing the fuel (1356.01 instead of 1518.28, 1358.04 and 1361.28 seconds) and less fuel (9.50 instead of 9.72, 9.53 and 9.51 *liters*). For example, regarding TD-Dijkstra-LTM-STTF, our savings in travel time are up to 0.38% combined with a small decrease in fuel consumption of up to 0.11%, leading to a reduction in the overall cost, on average, by 0.27% from 22.51 to 22.45 dollars.

We note that TD-Dijkstra-FSM produces coherent results with respect to the optimization criterion used. Thus, when the travel time criterion is used, the TT is effectively the lowest with respect to its value under the other optimization criteria. This pattern is not respected by the Dijkstra-SL,

TD-Dijkstra-LTM and TD-Dijkstra-LTM-STTF algorithms. For example, with the Dijkstra-SL the minimum cost (23.93) is obtained under the travel time optimization criterion. In addition, with both the TD-Dijkstra-LTM and TD-Dijkstra-LTM-STTF the best solutions for the travel time criterion are produced under the cost criterion for medium network instances. Finally, it is noticeable that our TD-Dijkstra-FSM algorithm provides the best solutions for all optimization criteria: 1,338.59 seconds for travel time, 9.34 liters for fuel consumption, and 22.45 dollars for costs. Table 4 clearly shows that in the presence of traffic congestion, using the TD-Dijkstra-FSM algorithm enhances the quality of solutions with respect to the TD-Dijkstra-LTM and TD-Dijkstra-LTM-STTF ones. Therefore, FSM improves solutions with high-frequency data. We can conclude that TD-Dijkstra-FSM is more efficient and realistic than the other ones.

Table 5: Comparative algorithm results considering the number of nodes scanning and labels

Optimization criteria	Instances	TD-Dijkstra-LTM		TD-Dijkstra-LTM-STTF		TD-Dijkstra-FSM	
		Nb. of nodes scanning	Nb. of labels	Nb. of nodes scanning	Nb. of labels	Nb. of nodes scanning	Nb. of labels
Travel Time	***L1-L20***	15517	33884	15567	33947	15283	33608
	M1-M20	8763	18877	8812	18916	8834	19185
	S1-S20	4873	10382	4846	10286	4788	10255
	D1-D20	19805	43878	19781	43762	19600	43812
	Average	12240	26755	12251	26728	**12126**	**26715**
Fuel Consumption	***L1-L20***	15234	33526	15267	33579	15207	33521
	M1-M20	8387	18234	8359	18182	8326	18116
	S1-S20	4309	9260	4324	9318	4337	9338
	D1-D20	20218	44993	20155	44880	20144	44939
	Average	12037	26503	12027	26490	**12004**	**26479**
Cost	***L1-L20***	15346	33573	15340	33515	15269	33403
	M1-M20	8554	18483	8618	18601	8624	18627
	S1-S20	4630	9898	4638	9905	4645	9930
	D1-D20	19919	44254	19945	44267	19901	44238
	Average	12112	26552	12135	26572	**12110**	**26550**

Additional experiments were conducted to assess the variations of cost and GHG emissions incurred as a consequence of traffic congestion during rush hours, such as at 16h00. Table 6 presents the experiments conducted to evaluate the impact of traffic congestion on the travel time, fuel consumption, and total cost. To this end, we now used the average results over the 60 instances (L*, M* and S*) with departure times ranging from 07h30 to 08h30, before the morning congestion, and ranging from 15h30 to 16h30 during the afternoon traffic. In the following, results of the time-independent Dijkstra-SL are not reported as it uses a fixed speed which is incoherent with this analysis.

Looking at the TD-Dijkstra-FSM with fuel consumption as the optimization criterion, we see that it increases from 6.37 to 6.81 *liters* (6.91%) when the path departure times are changed from 7h30 to 16h30. Similarly, the travel time increased, on average, by 14.66% induced by changes from 983.13 to 1,127.30 seconds. We observe the same pattern for the overall costs, which is increased, on average, by 11.10% induced by changes from 15.68 to 17.42 dollars. Overall, the TD-Dijkstra-FSM produced expected results with respect to traffic conditions. Further, when compared to the TD-Dijkstra-LTM and TD-Dijkstra-LTM-STTF algorithms, it produces coherent and consistent results with respect to the selected departure time and optimization criterion. Thus, when each optimization criterion is used, the corresponding metric (TT, Fuel or Costs) is effectively lower with respects to its value under the other optimization criteria. This pattern is not respected by the TD-Dijkstra-LTM as the minimum cost is obtained under the travel time optimization criterion for some departure times such as 07h30 (15.45 versus 15.49 dollars) and 08h00 (16.13 versus 16.38 dollars). Similarly, the TD-Dijkstra-LTM-STTF produces the minimum travel time under the cost optimization criterion for some departure times such as 07h30 (948.18 versus 947.27 seconds) and 15h30 (1028.02 versus 1027.95 seconds).

Figure 4 analyses in more details the impact of departure time on average path travel time and total cost for instances D1 to D20 (see Table 6). In Figure 4, the results of the TD-Dijkstra-FSM replicate the traffic pattern of Québec City with a moderate morning congestion between 7h30 and 9h00; low traffic between 10h00 and 14h30 which results in lower travel times and costs. Then, as expected, congestion rapidly increases between 15h00 and 15h30 to reach a peak between 16h00 and 17h30. Interestingly, allowing delayed or flexible departures can lead to better alternative paths yielding significant reduction of both GHG emissions and overall costs.

Table 7 shows further results when the cost minimization objective is used for the TD Lower Bound, TD Upper Bound, Dijkstra-SL, TD-Dijkstra-LTM, TD-Dijkstra-LTM-STTF and TD-Dijkstra-FSM algorithms. It shows that the TD-Dijkstra-LTM, TD-Dijkstra-LTM-STTF and TD-Dijkstra-FSM algorithms consistently provide average solution costs bounded by our lower and upper bounds. Indeed, for the TD-Dijkstra-FSM the gap between the lower bound value and the cost minimizing paths ranges from 3.77 to 7.07%, which are lower than those of the Dijkstra-SL, TD-Dijkstra-LTM and TD-Dijkstra-LTM-STTF ranging from 9.65 to 24.25%, 4.18 to 7.78%, and 4.37 to 8.34%, respectively. For example, when the departure time is 08h15 the results of the time-independent Dijkstra-SL for L* (29.68), M* (15.15) and S* (10.06) always exceed the corresponding upper

Table 6: Impacts of departure time

Departure Time	Optimization criteria	TD-Dijkstra-LTM				TD-Dijkstra-LTM-STTF				TD-Dijkstra-FSM			
		Avg Dist	Avg TT	Avg Fuel	Avg Cost	Avg Dist	Avg TT	Avg Fuel	Avg Cost	Avg Dist	Avg TT	Avg Fuel	Avg Cost
07h30	***Travel Time***	13695.50	938.67	6.57	15.45	13701.96	948.18	6.60	15.64	13663.22	935.43	6.55	15.48
	Fuel consumption	12754.75	989.78	6.38	15.75	12775.87	999.32	6.41	15.86	12762.90	983.13	6.37	15.68
	Cost	13449.36	943.28	6.50	15.49	13416.59	947.27	6.49	15.52	13410.70	939.78	6.47	15.43
07h45	***Travel Time***	14343.47	1005.68	6.96	16.51	13960.10	1008.02	6.84	16.43	14314.26	996.77	6.93	16.44
	Fuel consumption	12931.19	1028.83	6.54	16.26	12981.18	1034.67	6.57	16.35	12955.70	1022.32	6.53	16.20
	Cost	13316.18	1003.78	6.60	16.13	13284.40	1009.78	6.61	16.18	13307.89	997.72	6.59	16.05
08h00	***Travel Time***	13816.95	991.14	6.74	16.13	14299.76	1011.58	6.97	16.61	13776.51	986.06	6.72	16.15
	Fuel consumption	13074.91	1058.21	6.68	16.68	13054.92	1063.68	6.68	16.73	13083.87	1044.64	6.65	16.52
	Cost	13667.57	1012.77	6.76	16.38	13674.68	1015.40	6.77	16.42	13769.62	1004.20	6.77	16.32
08h15	***Travel Time***	14213.93	1044.37	7.04	16.95	14347.28	1042.70	7.08	17.00	14231.70	1030.20	7.00	16.81
	Fuel consumption	13029.15	1096.05	6.77	17.10	13050.57	1089.30	6.76	17.03	13041.42	1083.88	6.74	16.96
	Cost	13762.31	1042.22	6.89	16.78	13698.62	1044.68	6.87	16.78	13757.83	1037.08	6.87	16.72
08h30	***Travel Time***	14139.83	983.61	6.83	16.17	14259.28	996.28	6.90	16.41	14155.14	977.76	6.82	16.15
	Fuel consumption	13071.85	1035.03	6.60	16.39	13069.89	1042.13	6.63	16.48	13079.92	1031.15	6.59	16.34
	Cost	13653.33	994.92	6.69	16.15	113638.89	1000.27	6.71	16.21	13713.31	988.58	6.70	16.10
15h30	***Travel Time***	14296.88	1024.13	6.99	16.56	14301.36	1028.02	7.00	16.79	14205.08	1005.31	6.91	16.49
	Fuel consumption	13116.88	1058.94	6.70	16.70	13272.17	1057.00	6.73	16.73	13233.37	1038.92	6.67	16.51
	Cost	13595.96	1029.03	6.78	16.54	13584.94	1027.95	6.76	16.51	13541.31	1022	6.73	16.42
15h45	***Travel Time***	14355.48	1042.26	7.04	16.94	14436.86	1055.17	7.10	17.13	14302.52	1032.83	7.00	16.83
	Fuel consumption	13107.03	1093.23	6.78	17.09	13170.25	1095.45	6.80	17.13	13126.18	1088.15	6.77	17.03
	Cost	13706.53	1052.32	6.86	16.83	13750.48	1059.60	6.89	16.93	13824.59	1044.07	6.88	16.78
16h00	***Travel Time***	14387.18	1063.75	7.10	17.11	14643.84	1070.95	7.19	17.38	14501.99	1054.90	7.11	17.14
	Fuel consumption	13135.97	1122.06	6.84	17.40	13200.19	1113.12	6.84	17.33	13133.05	1107.13	6.81	17.24
	Cost	13666.95	1075.03	6.89	17.06	13669.11	1076.72	6.89	17.08	13720.86	1068.59	6.88	16.99
16h15	***Travel Time***	14425.41	1073.80	7.14	17.37	14466.19	1087.28	7.18	17.50	14320.00	1061.42	7.07	17.16
	Fuel consumption	13168.82	1128.38	6.87	17.50	13258.83	1131.40	6.90	17.55	13208.30	1117.80	6.85	17.38
	Cost	13878.132	1073.02	6.97	17.14	13885.51	1082.95	7.00	17.26	13926.41	1067.25	6.97	17.09
16h30	***Travel Time***	14529.90	1072.42	7.18	17.34	14589.84	1084.67	7.23	17.54	14374.23	1067.22	7.11	17.25
	Fuel consumption	12987.398	1141.72	6.85	17.58	13124.00	1138.68	6.87	17.58	13055.88	1127.30	6.81	17.42
	Cost	13879.06	1070.81	6.96	17.11	13568.04	1094.47	6.93	17.27	13924.58	1065.17	6.97	17.07

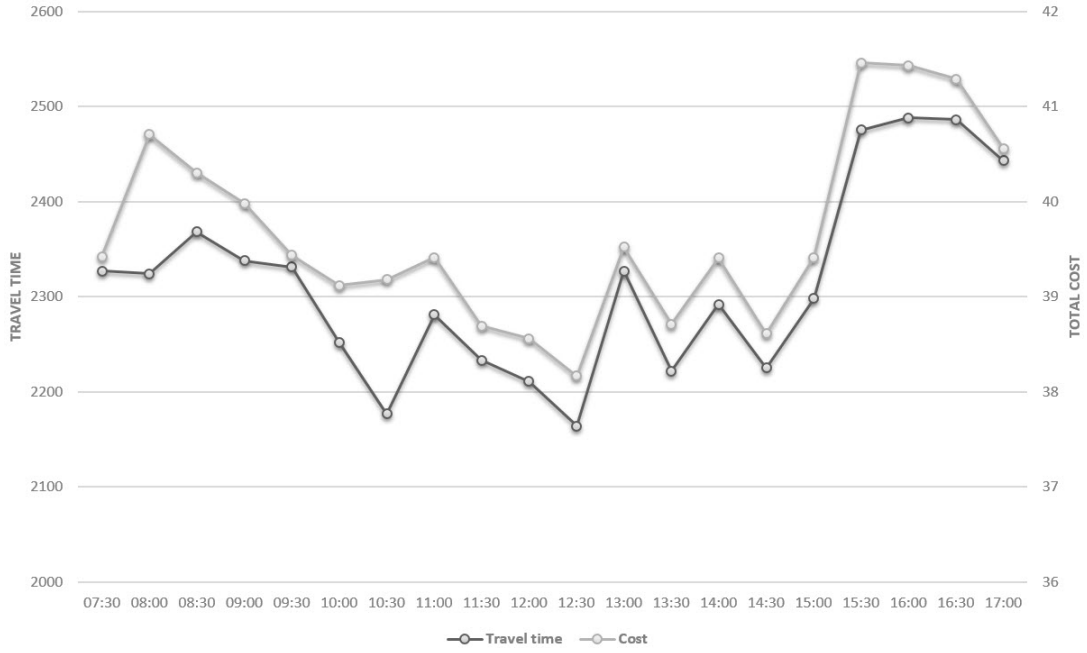

Figure 4: Impact of departure time

bounds of 29.22, 14.72 and 9.61 dollars showing that even our upper bound heuristic provides better results than this method. These results clearly show that the effectiveness of paths strongly increase if we consider time-varying speeds using TD-Dijkstra algorithms compared to those generated with Dijkstra-SL algorithm that uses fixed speeds. Further, we see that the proposed bounds capture the variation of traffic for each period according to the fluctuation of the travel cost ranging from 14.87 to 15.94$ for the lower bound and from 16.24 to 19.70$ for the upper bound.

Table 7: Average results under the total cost optimization criterion

Departure time	Instances	TD Lower Bounds		TD Upper Bounds		Dijkstra-SL			TD-Dijkstra-LTM			TD-Dijkstra-LTM-STTF			TD-Dijkstra-FSM		
		Cost	Sec	Cost	Sec	Cost	Gap (%)	Sec	Cost	Gap (%)	Sec	Cost	Gap (%)	Sec	Cost	Gap (%)	Sec
	L1-L20	25.90	0.30	28.21	0.20	28.40	9.64	0.20	27.17	4.88	0.56	27.11	4.67	0.77	27.03	4.33	0.43
	M1-M20	11.65	0.17	13.21	0.11	13.22	13.43	0.11	12.27	5.27	0.33	12.33	5.84	0.42	12.22	4.92	0.25
07h30	***S1-S20***	6.77	0.09	7.29	0.06	7.30	7.80	0.06	7.04	3.98	0.15	7.12	5.17	0.20	7.04	3.98	0.12
	Average	14.87	0.19	16.24	0.13	16.30	9.65	0.12	15.49	4.18	0.35	15.52	4.37	0.47	15.43	3.77	0.27
	L1-L20	26.03	0.29	28.58	0.19	28.87	10.92	0.19	27.35	5.10	0.55	27.42	5.34	0.79	27.24	4.67	0.42
	M1-M20	12.31	0.16	15.09	0.11	15.21	23.54	0.12	13.22	7.39	0.34	13.23	7.47	0.43	13.12	6.57	0.26
07h45	***S1-S20***	7.38	0.09	9.46	0.06	9.12	23.59	0.07	7.80	5.64	0.17	7.90	7.05	0.22	7.80	5.64	0.13
	Average	15.35	0.18	17.71	0.12	17.73	15.57	0.13	16.13	5.08	0.35	16.18	5.41	0.48	16.05	4.61	0.27
	L1-L20	26.15	0.29	29.22	0.19	29.68	13.52	0.19	27.68	5.85	0.54	27.65	5.74	0.77	27.53	5.29	0.41
	M1-M20	12.28	0.16	14.72	0.11	15.15	23.44	0.11	13.33	8.61	0.30	13.40	9.12	0.43	13.29	8.23	0.24
08h00	***S1-S20***	7.46	0.09	9.61	0.06	10.06	34.81	0.06	8.14	9.11	0.16	8.20	9.92	0.22	8.14	9.10	0.12
	Average	15.43	0.18	17.85	0.12	18.30	18.56	0.12	16.38	6.15	0.33	16.42	6.42	0.47	16.32	5.73	0.26
	L1-L20	26.25	0.29	28.95	0.19	29.26	11.48	0.19	27.70	5.55	0.55	27.69	5.49	0.78	27.59	5.14	0.42
	M1-M20	12.57	0.17	15.61	0.12	15.60	24.10	0.11	13.74	9.27	0.31	13.76	9.47	0.52	13.70	8.98	0.24
08h15	***S1-S20***	7.90	0.09	9.87	0.07	10.15	28.45	0.06	8.89	12.45	0.17	8.90	12.66	0.29	8.85	11.95	0.13
	Average	15.69	0.18	17.92	0.12	18.34	16.86	0.12	16.78	6.91	0.34	16.78	6.95	0.53	16.72	6.51	0.27
	L1-L20	25.94	0.31	28.37	0.21	28.60	10.24	0.19	27.32	5.31	0.55	27.33	5.36	0.79	27.20	4.85	0.42
	M1-M20	11.99	0.18	14.41	0.12	15.06	25.65	0.11	12.94	7.92	0.30	13.01	8.51	0.42	12.90	7.64	0.23
08h30	***S1-S20***	7.53	0.10	9.80	0.07	10.10	34.16	0.06	8.21	9.02	0.16	8.30	10.23	0.22	8.21	9.00	0.26
	Average	15.26	0.19	17.53	0.13	17.92	17.40	0.12	16.15	5.83	0.33	16.21	6.23	0.48	16.10	5.49	0.26
	L1-L20	26.08	0.30	29.56	0.20	29.38	12.66	0.19	27.49	5.42	0.55	27.41	5.10	0.78	27.33	4.80	0.43
	M1-M20	12.41	0.17	14.71	0.11	15.57	25.46	0.11	13.61	9.63	0.32	13.58	9.43	0.44	16.66	34.21	0.24
15h30	***S1-S20***	7.71	0.09	9.74	0.06	10.47	35.81	0.06	8.52	10.45	0.17	8.54	10.77	0.22	8.51	10.43	0.13
	Average	15.51	0.19	18.00	0.12	18.47	19.12	0.12	16.54	6.64	0.34	16.51	6.45	0.48	16.42	5.89	0.27
	L1-L20	26.60	0.30	30.19	0.20	30.09	13.10	0.20	28.28	6.29	0.57	28.20	6.02	0.78	28.15	5.80	0.44
	M1-M20	12.45	0.18	14.63	0.12	14.86	19.31	0.11	13.49	8.37	0.31	13.66	9.72	0.42	13.49	8.33	0.24
15h45	***S1-S20***	7.83	0.09	9.61	0.06	9.98	27.41	0.06	8.71	11.21	0.18	8.92	13.92	0.24	8.71	11.17	0.14
	Average	15.70	0.19	18.14	0.13	18.31	16.64	0.12	16.83	7.21	0.35	16.93	7.83	0.48	16.78	6.91	0.27
	L1-L20	26.81	0.30	30.78	0.20	30.84	15.05	0.20	28.53	6.40	0.58	28.46	6.15	0.86	28.46	6.17	0.45
	M1-M20	12.70	0.16	16.47	0.11	17.48	37.63	0.12	14.45	13.81	0.33	13.83	8.90	0.43	14.34	12.92	0.25
16h00	***S1-S20***	7.98	0.09	11.35	0.06	11.16	39.89	0.06	8.90	11.60	0.18	8.95	12.16	0.24	8.90	11.58	0.14
	Average	15.89	0.18	19.53	0.12	19.60	23.39	0.13	17.06	7.38	0.36	17.08	7.49	0.51	17.00	7.00	0.28
	L1-L20	26.88	0.30	31.90	0.20	31.86	18.52	0.20	28.59	6.35	0.59	28.78	7.07	0.79	28.52	6.11	0.45
	M1-M20	12.81	0.17	16.08	0.11	16.25	26.81	0.11	14.05	9.69	0.32	14.14	10.38	0.43	13.98	9.13	0.25
16h15	***S1-S20***	7.90	0.09	11.12	0.06	11.16	41.35	0.06	8.77	11.02	0.17	8.85	12.03	0.23	8.76	11.00	0.13
	Average	15.90	0.18	19.70	0.12	19.76	24.25	0.13	17.14	7.78	0.36	17.26	8.55	0.48	17.09	7.49	0.28
	L1-L20	27.19	0.29	31.80	0.19	31.86	17.19	0.20	28.59	5.15	0.59	29.05	6.84	0.80	28.52	4.92	0.45
	M1-M20	12.71	0.19	16.39	0.13	16.89	32.81	0.12	14.67	15.41	0.33	13.96	9.83	0.43	14.60	14.87	0.26
16h30	***S1-S20***	7.84	0.09	10.26	0.06	11.16	42.27	0.06	8.77	11.75	0.17	8.80	12.24	0.23	8.76	11.72	0.13
	Average	15.94	0.19	19.23	0.13	19.72	23.72	0.13	17.11	7.35	0.36	17.27	8.34	0.49	17.07	7.07	0.28

The final set of experiments presented in Table 8 aims at providing some insight on the impact of carried loads over our four performance measures. Results are obtained under the total cost minimization criteria and are averages over all instances. As expected, as the load increases, fuel consumption, and thus the cost, increase for the TD-Dijkstra algorithms. We can see that the paths are updated (as Distance and Travel Time change) when the load increases from 15 to 20 tons. However, the paths remain the same with increased fuel consumption when the load increases from 20 to 25 tons in the case of the TD-Dijkstra-LTM. We note that the same pattern holds for the TD-Dijkstra-LTM-STTF from 22.5 to 25 tons. For the TD-Dijkstra-FSM, the fuel consumption increases from 7.87 to 8.27 *liters* (5.08%) from 15 to 17.50 tons, from 8.27 to 8.68 *liters* (4.96%) from 17.5 to 20 tons, from 8.68 to 9.08 *liters* (4.61%) from 20 to 22.5 tons, and from 9.09 to 9.50 *liters* (4.51%) from 22.5 to 25 tons. Therefore, the fuel consumption increases slowly and the total distance is slightly decreased from empty load (15 tons) to TL (25 tons), thus having a slightly broader impact on the obtained path according each load pattern. In addition, we have noticed that again the TD-Dijkstra-FSM outperforms the other ones for the different load patterns.

Table 8: Impact of carried load on performance measures

Algorithms	Performance measures	Carried Load				
		Empty Load (15 t)	LTL (17.5 t)	LTL (20 t)	LTL (22.5 t)	TL (25 t)
TD-Dijkstra-LTM	Avg Dist	19657.96	19573.35	19610.46	19610.46	19610.46
	Avg TT	1356.21	1353.25	1358.04	1358.04	1358.04
	Avg Fuel	7.89	8.27	8.71	9.12	9.53
	Avg Cost	20.60	21.02	21.56	22.03	22.50
TD-Dijkstra-LTM-STTF	Avg Dist	19633.30	19575.45	19560.31	19528.51	19528.51
	Avg TT	1357.11	1358.30	1359.11	1361.28	1361.28
	Avg Fuel	7.88	8.28	8.69	9.10	9.51
	Avg Cost	20.60	21.07	21.55	22.03	22.51
TD-Dijkstra-FSM	Avg Dist	19642.42	19595.47	19613.83	19578.88	19574.87
	Avg TT	1353.80	1349.91	1351.34	1351.81	1351.91
	Avg Fuel	7.87	8.27	8.68	9.08	9.50
	Avg Cost	20.56	20.98	21.47	21.94	22.41

7 Conclusions

The TDMCP-SO extends the TDQPP by considering fuel consumption/GHG emissions minimization. This extension is of high practical relevance since traffic congestion is an important issue for

logistics providers in urban contexts. Accurate and tight time-dependent least cost lower and upper bounds were derived based on MCP-SO properties. A fast and effective time-dependent Dijkstra label-setting algorithm and a lower bounding method have been implemented for 80 benchmark instances based on a large road network in Québec City including more than 17000 nodes and 24 million speed observations. The designed algorithms combine pre-existing CMEM and FSM models to compute GHG emissions and costs using time-varying speeds. Our algorithm is highly effective in finding good-quality solutions for benchmark instances of all sizes using a FIFO-consistent method overcoming the challenge of the non-linearities of emissions.

The extensive computational experiments demonstrated the benefit of choosing alternative paths in congested urban areas under FIFO consistency that leads to substantial fuel consumption/GHG emissions reduction and cost savings. We clearly demonstrate that using our time-dependent FIFO-consistent method reduces the computational time and leads to better results with respect to the ones which use constant speeds, LTM and smoothing methods. Moreover, it produces coherent and consistent results with respect to the departure time, optimization criterion used, and load patterns. An interesting insight derived from this research is that avoiding traffic congestion during peak hours yields substantial GHG emissions reductions and costs savings. Our time-dependent models reproduce expected and coherent behavior with respect to optimization criteria, time of the day (level of congestion), carried loads and selected paths which is not the case for the existing methods. We have also shown that carried loads slightly affect the chosen path, particularly as the vehicle load becomes larger, the potential savings in fuel consumption and GHG emissions also increase.

Further research should consider how to embed TDMCP-SO algorithms and our lower bounding method into local search heuristics to efficiently solve real-world time-dependent distribution problems considering emissions minimization based on time-varying speeds. Adding time-dependent quickest path optimization may enhance the resulting route plans that are selected based on dynamic paths to avoid traffic congestion across real road networks. Indeed, the studied TDMCP-SO could be integrated into TDVRPs and the proposed time-dependent bounds could be used to design accurate bounds and develop efficient heuristics such as goal-directed search and neighborhood search heuristics to solve large-scale instance of TDVRP involving the determination of cost-optimal routes. Regarding the TDVRP over stochastic and time-dependent network, methods such as polynomial approximation would seem appropriate to adapt our bounds to handle speed stochasticity

and find optimal routing policy.

References

[1] Matthew Barth and Kanok Boriboonsomsin. Real-world carbon dioxide impacts of traffic congestion. *Transportation Research Record: Journal of the Transportation Research Board*, 2058:163–171, 2008.

[2] Matthew Barth and Kanok Boriboonsomsin. Energy and emissions impacts of a freeway-based dynamic eco-driving system. *Transportation Research Part D: Transport and Environment*, 14 (6):400–410, 2009.

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

[4] Khaled Belhassine, Leandro C Coelho, Jacques Renaud, and Jean-Philippe Gagliardi. Improved home deliveries in congested areas using geospatial technology. Technical Report CIRRELT-2018-02, Montreal, Canada, 2018.

[5] Gerth Stølting Brodal and Riko Jacob. Time-dependent networks as models to achieve fast exact time-table queries. *Electronic Notes in Theoretical Computer Science*, 92:3–15, 2004.

[6] Tobia Calogiuri, Gianpaolo Ghiani, and Emanuela Guerriero. The time-dependent quickest path problem: Properties and bounds. *Networks*, 66(2):112–117, 2015.

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

[8] Said Dabia, Emrah Demir, and Tom Van Woensel. An exact approach for a variant of the pollution-routing problem. *Transportation Science*, 51(2):607–628, 2017.

[9] Brian C Dean. Algorithms for minimum-cost paths in time-dependent networks with waiting policies. *Networks*, 44(1):41–46, 2004.

[10] Brian C Dean. Shortest paths in FIFO time-dependent networks: Theory and algorithms. Technical report, Massachusetts Institute of Technology, 2004.

[11] Frank Dehne, Masoud T Omran, and Jörg-Rüdiger Sack. Shortest paths in time-dependent FIFO networks. *Algorithmica*, 62(1-2):416–435, 2012.

[12] Daniel Delling and Giacomo Nannicini. Core routing on dynamic time-dependent road networks. *INFORMS Journal on Computing*, 24(2):187–201, 2012.

[13] Emrah Demir, Tolga Bektaş, and Gilbert Laporte. An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223 (2):346–359, 2012.

[14] Emrah Demir, Tolga Bektaş, and Gilbert Laporte. The bi-objective pollution-routing problem. *European Journal of Operational Research*, 232(3):464–478, 2014.

[15] Emrah Demir, Tolga Bektaş, and Gilbert Laporte. A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793, 2014.

[16] Marco Di Bartolomeo, Enrico Grande, Gaia Nicosia, and Andrea Pacifici. Cheapest paths in dynamic networks. *Networks*, 55(2):23–32, 2017.

[17] Jan Fabian Ehmke, Stephan Meisel, and Dirk Christian Mattfeld. Floating car based travel times for city logistics. *Transportation Research Part C: Emerging Technologies*, 21(1):338–352, 2012.

[18] Jan Fabian Ehmke, Ann Melissa Campbell, and Barrett W Thomas. Data-driven approaches for emissions-minimized paths in urban areas. *Computers & Operations Research*, 67:34–47, 2016.

[19] Jan Fabian Ehmke, Ann Melissa Campbell, and Barrett W Thomas. Vehicle routing to minimize time-dependent emissions in urban areas. *European Journal of Operational Research*, 251(2):478–494, 2016.

[20] Jan Fabian Ehmke, Ann M Campbell, and Barrett W Thomas. Optimizing for total costs in vehicle routing in urban areas. *Transportation Research Part E: Logistics and Transportation Review*, 116:242–265, 2018.

[21] Bernhard Fleischmann, Martin Gietz, and Stefan Gnutzmann. Time-varying travel times in vehicle routing. *Transportation Science*, 38(2):160–173, 2004.

[22] Anna Franceschetti, Dorothée Honhon, Tom Van Woensel, Tolga Bektaş, and Gilbert Laporte. The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological*, 56:265–293, 2013.

[23] Anna Franceschetti, Emrah Demir, Dorothée Honhon, Tom Van Woensel, Gilbert Laporte, and Mark Stobbe. A metaheuristic for the time-dependent pollution-routing problem. *European Journal of Operational Research*, 259(3):972–991, 2017.

[24] Song Gao and Ismail Chabini. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B: Methodological*, 40(2):93–122, 2006.

[25] Song Gao and He Huang. Real-time traveler information for optimal adaptive routing in stochastic time-dependent networks. *Transportation Research Part C: Emerging Technologies*, 21(1):196–213, 2012.

[26] Gianpaolo Ghiani and Emanuela Guerriero. A lower bound for the quickest path problem. *Computers & Operations Research*, 50:154–160, 2014.

[27] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2): 100–107, 1968.

[28] John Hickman, Dieter Hassel, Robert Joumard, Zissis Samaras, and S Sorenson. *Methodology for calculating transport emissions and energy consumption*. 1999. URL `http://www.transport-research.info/sites/default/files/project/documents/meet.pdf`. Available online (accessed on May 21, 2018).

[29] He Huang and Song Gao. Optimal paths in dynamic networks with dependent random link travel times. *Transportation Research Part B: Methodological*, 46(5):579–598, 2012.

[30] Yixiao Huang, Lei Zhao, Tom Van Woensel, and Jean-Philippe Gross. Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological*, 95:169–195, 2017.

[31] Soumia Ichoua, Michel Gendreau, and Jean Yves Potvin. Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research*, 144(2):379–396, 2003.

[32] O Jabali, T Van Woensel, and A.-G. de Kok. Analysis of travel times and CO_2 emissions in time-dependent vehicle routing. *Production and Operations Management*, 21(6):1060–1074, 2012.

[33] Adrianus Leendert Kok, EW Hans, and JMJ Schutten. Vehicle routing under time-dependent travel times: the impact of congestion avoidance. *Computers & Operations Research*, 39(5): 910–918, 2012.

[34] Elise Miller-Hooks and Baiyu Yang. Updating paths in time-varying networks given arc weight changes. *Transportation Science*, 39(4):451–464, 2005.

[35] Ministère de l'Énergie et des Ressources Naturelles. Facteurs d'émission et de conversion. Bureau de l'Efficacité et de l'Innovation Énergétique, Gouvernement du Québec. Available online (accessed on May 21, 2018), 2014. URL `http://www.transitionenergetique.gouv.qc.ca/fileadmin/medias/pdf/Facteurs_emissions.pdf`.

[36] Ariel Orda and Raphael Rom. Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the Association for Computing Machinery*, 37(3): 607–625, 1990.

[37] Jiani Qian and Richard Eglese. Fuel emissions optimization in vehicle routing problems with time-varying speeds. *European Journal of Operational Research*, 248(3):840–848, 2016.

[38] Martin Savelsbergh and Tom Van Woensel. City logistics: Challenges and opportunities. *Transportation Science*, 50(2):579–590, 2016.

[39] Hanif D Sherali, Antoine G Hobeika, and Sasikul Kangwalklai. Time-dependent, label-constrained shortest path problems with applications. *Transportation Science*, 37(3):278–293, 2003.

[40] Yunchuan Sun, Xinpei Yu, Rongfang Bie, and Houbing Song. Discovering time-dependent shortest path on traffic graph for drivers towards green driving. *Journal of Network and Computer Applications*, 83:204–212, 2017. ISSN 1084-8045.

[41] Kiseok Sung, Michael GH Bell, Myeongki Seong, and Soondal Park. Shortest paths in a network with time-dependent flow speeds. *European Journal of Operational Research*, 121(1): 32–39, 2000.

[42] Transports Canada. Transportation in Canada 2016. Available online (accessed on May 21, 2018), 2017. URL `https://www.tc.gc.ca/media/documents/policy/comprehensive_report_2016.pdf`.

[43] Marcel Turkensteen. The accuracy of carbon emission and fuel consumption computations in green vehicle routing. *European Journal of Operational Research*, 262(2):647–659, 2017.

[44] Tom Van Woensel, Laoucine Kerbache, Herbert Peremans, and Nico Vandaele. Vehicle routing with dynamic travel times: A queueing approach. *European Journal of Operational Research*, 186(3):990–1007, 2008.

[45] Liang Wen and Richard Eglese. Minimum cost VRP with time-dependent speed data and congestion charge. *Computers & Operations Research*, 56:41–50, 2015.

[46] Liang Wen, Bülent Çatay, and Richard Eglese. Finding a minimum cost path between a pair of nodes in a time-varying road network with a congestion charge. *European Journal of Operational Research*, 236(3):915–923, 2014.

[47] Lixing Yang and Xuesong Zhou. Constraint reformulation and a lagrangian relaxation-based solution algorithm for a least expected time path problem. *Transportation Research Part B: Methodological*, 59:22–44, 2014.