



CIRRELT

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

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Time-Dependent Quickest Path Problem with Emission Minimization

Hamza Heni
Leandro C. Coelho
Jacques Renaud

October 2017

CIRRELT-2017-62

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval,
sous le numéro FSA-2017-014.

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

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

www.cirrelt.ca

Time-Dependent Quickest Path Problem with Emission Minimization

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 de la Terrasse, Université Laval, Québec, Canada G1V 0A6

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 the time-dependent quickest path problem with emission minimization (TDQPP-EM), 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 using three adaptation of the Dijkstra's label-setting algorithm. We show that considering time-varying speeds provides substantial improvements over the one based only on fixed speeds. Our computational results demonstrate that contrary to the TDQPP, the TDQPP-EM is more difficult to solve to optimality but the proposed algorithms are shown to be robust and efficient in reducing the total cost even for large instances.

Keywords. Time-dependent networks, congestion, emission, quickest path, label-setting algorithm.

Acknowledgements. 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 Ministère de l'Énergie et des Ressources naturelles, government of Québec. These supports are gratefully acknowledged. We also thank our industrial partners for their outstanding collaboration, availability and for providing the data.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

* Corresponding author: Leandro.Coelho@cirrelt.ca

1 Introduction

Road freight transportation is a significant contributor to greenhouse gas (GHG) emissions [11]. This is mostly driven by increased traffic congestion due to the high number of vehicles on urban areas. When traveling on cities, fuel consumption and GHG emissions are highly affected by speed levels depending on 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 at both operational and environmental dimensions [28, 16].

Many works have demonstrated the importance of speed in minimizing travel costs. 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. In this work we introduce and solve a new variant of both the classical time-dependent quickest path problem (TDQPP) of Calogiuri et al. [4] and the time-dependent emissions-minimizing path problem of Ehmke et al. [13] which we call the Time-Dependent Quickest Path Problem with Emission Minimization (TDQPP-EM). 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 overall objective is to determine the least cost path based on a time-dependent network with time-varying speeds. 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 our comprehensive objective function, and we compute lower and upper bounds on the overall cost taking into account the time-dependent context.

By conducting extensive experiments, we show that GHG emissions, fuel consumption as well as cost savings can be achieved through the fast computation of point-to-point least cost paths using a comprehensive objective function composed of GHG emissions, fuel and driver costs, instead of the traditional distance-minimizing and least time path objectives of the TDQPP.

This paper makes several important contributions to the literature:

- we introduce the TDQPP-EM by explicitly considering GHG emissions and fuel consumption as parts of the cost components to capture both environmental and operational impacts;
- we prove that, under the first-in, first-out (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 factors 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 dynamic. This ensures that our solution methods account for the impact of speed variations on the optimization of a chosen path;
- finally, we adapt Dijkstra’s time-dependent label-setting algorithm to solve the TDQPP-EM.

The reminder 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 TDQPP-EM, and present some properties of the TDQPP-EM. 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. Our conclusions are presented in Section 7.

2 Literature review

The TDQPP-EM is a problem in the field of green road freight transportation [11], 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 [10, 16].

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) [8, 4], least emissions path (LEP) [13, 14] and least cost path (LCP) [12]. The TDQPP-EM is an extension of the TDQPP considering a time-dependent travel cost and can be seen as a variant of the shortest path problem (SPP) over time-dependent networks, which is $\mathcal{NP}$ -hard.

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 paths 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 [5]. 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 dynamics the TDQPP can be solved optimally and efficiently in polynomial time by adapting any label-setting shortest path algorithm [7].

Moreover, many existing works have not explicitly considered whether the FIFO assumption holds [24, 13], making shortest path algorithms very unstable. Sherali et al. [29] 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 dynamics, Sung et al. [30] proposed the flow speed model (FSM) in which the flow speed of each arc depends on the time period. 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 rule as the arc travel time changes as the period changes. The determination of quickest paths with LTM requires some additional steps to ensure the FIFO dynamics by eliminating potential cycles if waiting at nodes is not allowed [24] and deriving a travel time function satisfying the non-passing consistency [15]. Recently, Yang and Zhou [35] 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. [21] proposed a time-dependent speed model that respects the FIFO dynamics. 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. [32] proposed a queuing approach to capture traffic congestion and model travel times. A study by Kok et al. [23] 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 [19]. 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.

Ghiani and Guerriero [18] proposed an effective lower bound for the quickest path problem, which was embedded into an A* algorithm. Calogiuri et al. [4] studied the properties and bounds of TDQPP. Using the time-dependent speed model of Ichoua et al. [21], they prove that under the FIFO assumption, 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

Bektaş and Laporte [3] introduced the pollution-routing problem. Based on the comprehensive modal emissions model (CMEM) (see Section 2.3) of Barth and Boriboonsomsin [1], 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. [9] extend it by applying a speed optimization algorithm, identi-

fying the optimal speed on each arc in order to minimize the expected costs of fuel consumption and driver wages.

Focusing on the analysis of time-dependent costs as a function of speed, load and fuel consumption, Franceschetti et al. [17] extended the PRP to a time-dependent setting using the time-dependent travel time model of Jabali et al. [22]. Recently, Franceschetti et al. [16] developed a metaheuristic approach to solve the PRP under congestion, which integrates departure time and speed optimization procedures.

In Wen and Eglese [34], the authors solve the vehicle routing problem (VRP) with time-dependent speeds, where the total cost involves fuel cost, driver cost and congestion charge. In their work, the impact of vehicle load is not considered, and fixed congestion charges are applied once per day for each vehicle.

The results of the previous works show that the traditional objectives consisting of only 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. Indeed, there is a gap in the PRP research area related to the integration of GHG emissions models into TDQPP. Few papers have addressed path flexibility and GHG emissions-minimized paths. The exceptions are the works of Wen et al. [33], Ehmke et al. [13, 14], Qian and Eglese [26] and Huang et al. [20]. Following these works, we assume that vehicles must travel at the speed of traffic and do not have the ability to control their speed in a way that minimizes costs. Additionally, we consider that in time-dependent networks, congestion patterns are variable across each segment. Our goal is to model a more comprehensive objective function which captures and minimizes the cost of GHG emissions and fuel consumption along with operational costs considering time-varying speeds on the underlying road network.

2.3 Comprehensive modal emissions model

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:

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

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 $\alpha = a + g \sin \theta + gC_r \cos \theta$, $\beta = 0.5C_d A \rho$, $\varsigma = \frac{1}{(1000\varepsilon\eta_{tf})}$ and $\lambda = \frac{\zeta}{\varpi\psi}$, and based on the assumption associated with values of used parameters, expression (1) can be rewritten as:

$$e_r = \lambda(kN_e V + \varsigma\alpha(w+q)s + \varsigma\beta s^3). \quad (2)$$

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 (kg/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)	11.111
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

3 Formal description and problem statement

In this section, we introduce our notation, give a formal definition for the TDQPP-EM 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 $\mathcal{Z} = \{t_0, t_0 + \delta, \dots, t_0 + H\delta\}$, 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 $h = 0, 1, 2, \dots, H - 1$. 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 . Hence, the travel speed of each arc is assumed to be dynamic for a particular traveler across any arc. We denote the distance between two nodes i, j as L_{ij} . 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 $\tau : \mathcal{A} \rightarrow \mathbb{R}^+$ and $c : \mathcal{A} \rightarrow \mathbb{R}^+$ 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_0 the TDQPP-EM aims to determine a path $p = (o =$

$v_0, \dots, v_i, \dots, v_j, \dots, v_k = d$) such that the time-dependent total cost $\varphi_p(t_0)$ between source $o \in \mathcal{V}$ and destination $d \in \mathcal{V}$ is minimum.

Following the FSM, we assume that speed s_{ij}^h on arc (i, j) is constant over a time slot:

$$s_{ij}^h = \sigma_{ijh} u_{ij}, \quad (3)$$

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 t at node i and the arrival time $\gamma_j^p(t)$ at node j belong to, respectively, with $h_t \in \{0, \dots, H-1\}$ and $h_\gamma \in \{h_t, \dots, H-1\}$. The travel time along an arc is the sum of three portions of time:

- (i) The time associated with the first time interval h_t : $z_{h_t+1} - \gamma_i^p(t)$.
- (ii) The duration of the $(h_\gamma - h_t - 2)$ intermediate time intervals crossed when travelling along arc (i, j) : $\sum_{h=h_t+1}^{h_\gamma-1} (z_{h+1} - z_h)$.
- (iii) The time related to the last time slot: $\gamma_j^p(t) - z_{h_\gamma}$.

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

$$\tau_{ij}(t) = \begin{cases} L_{ij}/(\sigma_{ijh_t} u_{ij}) & \text{if } h_\gamma = h_t \\ \gamma_j^p(t) - \gamma_i^p(t) & \text{if } h_\gamma = h_t + 1 \\ (z_{h_t+1} - \gamma_i^p(t)) + (h_\gamma - h_t - 1) \delta + (\gamma_j^p(t) - z_{h_\gamma}) & \text{if } h_\gamma > h_t + 1. \end{cases} \quad (4)$$

The arrival time $\gamma_{v_i}^p$ at node v_i is expressed as follows (we omit the indices $v_{i-1}v_i$ on variables L, l , and s to simplify the notation):

$$\gamma_{v_i}^p(t) = \begin{cases} L/s^{h-1} + t & \text{if } L/s^h < z_h - t, h = h_t \\ (L - l^{h-1})/s^h + z_h & \text{if } (L - l^{h-1})/s^h < z_{h+1} - z_h, h \in \{h_t + 1, \dots, h_\gamma\}, \end{cases} \quad (5)$$

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 $h \in \{h_t + 1, \dots, h_\gamma\}$, and $L_{ij} = \sum_{h=h_t}^{h_\gamma} l^h$. Note that the traversal time $\Gamma_p(t)$ of a path $p = (o = v_0, v_2, \dots, v_k = d)$ can be induced from the arrival time at the destination node d . Therefore, its given by:

$$\Gamma_p(t) = \gamma_{v_d}^p(t) - t. \quad (6)$$

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, namely Bektaş and Laporte [3], Demir et al. [9], Franceschetti et al. [17, 16], Dabia et al. [6] and Huang et al. [20], among others. According to these works GHG emissions are directly proportional to fuel consumption.

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 (2) and (4):

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

where

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

and

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

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

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

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 (6):

$$\varphi(\Gamma_p(t)) = c_d \Gamma_p(t). \quad (11)$$

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:

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

As we know that 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:

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

where

$$c_{ij}(\gamma_i^p(t)) = c_d \tau_{ij}(\gamma_i^p(t)) + c_f \mathcal{F}_{ij}(\gamma_i^p(t)). \quad (14)$$

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

A mathematical programming formulation for the TDQPP-EM is presented in Appendix A.

4 Time-dependent lower and upper bounds for the TDQPP-EM

This section presents time-dependent lower and upper bounds for the TDQPP-EM that can be computed by ignoring the network-wide traffic congestion. Let $\mathcal{P}_\varphi$ be the set of all feasible paths of the TDQPP-EM on $\mathcal{G}$. Given a path $p \in \mathcal{P}_\varphi$, let $\Gamma(p)$ be the traversal time of p assuming (3) holds. We also denote by $\underline{\Gamma}(p)$ the traversal time of p if the congestion ratios of all arcs are set to their heaviest values $\sigma_h = \min_{ijh}(\sigma_{ijh})$ for each time slot Z_h . 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 and the TDQPP-EM are reduced to the QPP and to the QPP-EM, respectively. 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_0 . Let p_c^* and $\underline{\underline{p}}_c^*$ be optimal solutions of the TDQPP-EM and QPP-EM, 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 TDQPP-EM under the speed variation relationship (3) presents multiple advantages. Firstly, efficient algorithms designed for the QPP can be immediately applied to solve the TDQPP-EM. 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 TDQPP-EM. Indeed, in the following subsections we prove that $\varphi(\underline{\underline{\Gamma}}(p^*)) + \varphi(\mathcal{F}(p^*, q_0, s^*)) \leq \varphi(p_c^*)$ is a lower bound on $\varphi(p_c^*)$ and that $\min\{\varphi(\underline{\underline{p}}_c^*), \varphi(\underline{\underline{p}}^*), \varphi(\underline{p}^*), \varphi(p^*)\} \leq \varphi(\underline{\underline{p}}^*)$ is its upper bound:

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

where $q_0 = 0$ and $s^* = (\frac{kNV}{2\beta\varsigma})^{1/3}$ is the optimal speed which minimizes 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 $\varphi(\underline{\underline{\Gamma}}(p^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*)) \leq \varphi(p_c^*)$ is a lower bound on $\varphi(p_c^*)$ and that if the vehicle travels with the same load at speed s^* that minimizes fuel consumption across all arcs, then $\underline{\underline{p}}^*$ is optimal for TDQPP-EM, that is, $\underline{\underline{p}}^* = \underline{\underline{p}}_c^*$.

Theorem 1 *Path $\underline{\underline{p}}^*$ is an optimal solution for the TDQPP-EM 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.*

The proof of Theorem 1 is provided in Appendix B, Proof 1.

Corollary 1.1 *Given two optimal paths $\underline{\underline{p}}^*$ and $\underline{\underline{p}}_c^*$ (with respect to $\varphi(\underline{\underline{p}}^*)$ and $\varphi(\underline{\underline{p}}_c^*)$, respectively) for the TDQPP and the TDQPP-EM, respectively, the following relationship is satisfied:*

$$\varphi(\Gamma(\underline{\underline{p}}^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*)) \leq \varphi(\underline{\underline{p}}_c^*). \quad (16)$$

The proof of Corollary 1.1 is included in Appendix B, Proof 2.

4.2 A worst case analysis

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

Theorem 2 *The value $\varphi(\underline{\underline{p}}_c^*)$ is an upper bound not greater than $\frac{1}{\Delta}\varphi(\Gamma(\underline{\underline{p}}^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*))$.*

The proof of Theorem 2 is given by Proof 3 in Appendix B.

5 Design of efficient TDQPP-EM algorithms

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

5.1 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 time and travel time across arc (i, j) can be computed through Algorithm 1.

In the case of the LTM the travel time of arc (i, j) is specified when departing from node i at 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 2.

5.2 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 3 and 4, respectively.

Algorithm 1 Computing the travel time $\tau_{ij}(\gamma_i^p(t))$ across a given arc (i, j) based on the FSM

```

1: function TRAVEL_TIME_FSM( $\gamma_i^p(t)$ ,  $(i, j)$ ,  $\mathcal{Z}$ ,  $\mathcal{S}$ )
2:    $h | \gamma_i^p(t) \in Z_h = [z_h, z_{h+1}[$ 
3:    $k \leftarrow h$ 
4:    $d \leftarrow L_{ij} - [s_{ij}^k(z_{k+1} - \gamma_i^p(t))]$ 
5:   while  $d > 0$  do
6:      $k \leftarrow k + 1$ 
7:      $d \leftarrow d - [s_{ij}^k(z_{k+1} - z_k)]$ 
8:   end while
9:    $\gamma_j^p(t) \leftarrow z_{k+1} + d/s_{ij}^k$ 
10:   $\tau_{ij}(\gamma_i^p(t)) \leftarrow \gamma_j^p(t) - \gamma_i^p(t)$ 
    return  $\tau_{ij}(\gamma_i^p(t))$ 
11: end function

```

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

```

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

```

In Algorithm 3, 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.

5.3 Time-dependent Dijkstra algorithms

In this section we propose new solution methods based on two adaptation of Dijkstra's label-setting algorithm. The travel cost across each arc is computed according to the set of flow speeds obtained at the time of traversing the arc. Therefore, FSM and LTM models for the computation of time-dependent arc arrival time 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 and LTM algorithms.

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

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

```

1: function TRAVEL_COST_FSM( $\gamma_i^p(t)$ ,  $(i, j)$ ,  $\mathcal{Z}$ ,  $\mathcal{S}$ )
2:    $h | \gamma_i^p(t) \in Z_h = [z_h, z_{h+1}[$ 
3:    $k \leftarrow h$ 
4:    $l \leftarrow s_{ij}^k(z_{k+1} - \gamma_i^p(t))$ 
5:    $d \leftarrow L_{ij} - l$ 
6:    $g \leftarrow \lambda k N_e V \left( \frac{l}{s_{ij}^k} \right) + l \lambda \varsigma \beta (s_{ij}^k)^2$ 
7:   while  $d > 0$  do
8:      $k \leftarrow k + 1$ 
9:      $l \leftarrow s_{ij}^k(z_{k+1} - z_k)$ 
10:     $g \leftarrow g + \lambda k N_e V \left( \frac{l}{s_{ij}^k} \right) + l \lambda \varsigma \beta (s_{ij}^k)^2$ 
11:     $d \leftarrow d - [s_{ij}^k(z_{k+1} - z_k)]$ 
12:  end while
13:   $\gamma_j^p(t) \leftarrow z_{k+1} + d/s_{ij}^k$ 
14:  if  $k > h$  then
15:     $l \leftarrow s_{ij}^k(\gamma_j^p(t) - z_k)$ 
16:     $g \leftarrow g + \lambda k N_e V \left( \frac{l}{s_{ij}^k} \right) + l \lambda \varsigma \beta (s_{ij}^k)^2$ 
17:  else
18:     $g \leftarrow \lambda k N_e V \left( \frac{L_{ij}}{s_{ij}^h} \right) + L_{ij} \lambda \varsigma \beta (s_{ij}^h)^2$ 
19:  end if
20:   $\tau_{ij}(\gamma_i^p(t)) \leftarrow \gamma_j^p(t) - \gamma_i^p(t)$ 
21:   $\mathcal{F}_{ij}(\gamma_i^p(t)) \leftarrow \lambda \varsigma \alpha (w + q) L_{ij} + g$ 
22:   $c_{ij}(\gamma_i^p(t)) \leftarrow c_d \tau_{ij}(\gamma_i^p(t)) + c_f \mathcal{F}_{ij}(\gamma_i^p(t))$ 
    return  $c_{ij}(\gamma_i^p(t))$ 
23: end function

```

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

```

1: function TRAVEL_COST_LTM( $\gamma_i^p(t)$ ,  $(i, j)$ ,  $\mathcal{Z}$ ,  $\mathcal{S}$ )
2:    $h | \gamma_i^p(t) \in Z_h = [z_h, z_{h+1}[$ 
3:    $\tau_{ij}(\gamma_i^p(t)) \leftarrow s_{ij}^h / L_{ij}$ 
4:    $\gamma_j^p(t) = \gamma_i^p(t) + \tau_{ij}(\gamma_i^p(t))$ 
5:    $\mathcal{F}_{ij}(\gamma_i^p(t)) \leftarrow \lambda (k N_e V + \varsigma \alpha (w + q) s_{ij}^h + \varsigma \beta (s_{ij}^h)^3) \frac{L_{ij}}{s_{ij}^h}$ 
6:    $c_{ij}(\gamma_i^p(t)) \leftarrow c_d \tau_{ij}(\gamma_i^p(t)) + c_f \mathcal{F}_{ij}(\gamma_i^p(t))$ 
    return  $c_{ij}(\gamma_i^p(t))$ 
7: end function

```

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.

Let $Travel_Cost_*(\cdot)$ 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 or LTM). 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 5.

Algorithm 5 Determination of a near-optimal least cost path by adapting Dijkstra label-setting algorithm (TD-Dijkstra-*)

```

1: function TD_DIJKSTRA( $o, d, \mathcal{G}$ )
2:    $\mathcal{E} \leftarrow \emptyset$ 
3:    $\mathcal{N} \leftarrow \mathcal{V}$ 
4:    $c_i \leftarrow \infty, \forall i \in \mathcal{V}$ 
5:    $c_o \leftarrow 0$  and  $predecessor(o) \leftarrow o$ 
6:   while  $|\mathcal{E}| < n$  do
7:     let  $i \in \mathcal{N}$  be a node for which  $c_i \leftarrow \min\{c_j : j \in \mathcal{N}\}$ 
8:      $\mathcal{E} \leftarrow \mathcal{E} \cup \{i\}$ 
9:      $\mathcal{N} \leftarrow \mathcal{N} \setminus \{i\}$ 
10:    if  $i = d$  then
11:      Stop
12:    end if
13:    for each  $(i, j) \in \mathcal{E}^+(i)$  do
14:      if  $c_j > Travel\_Cost\_*(\gamma_i^p(t), (i, j), \mathcal{Z}, \mathcal{S})$  then
15:         $c_j \leftarrow c_i + Travel\_Cost\_*(\gamma_i^p(t), (i, j), \mathcal{Z}, \mathcal{S})$ 
16:         $predecessor(j) \leftarrow i$ 
17:      end if
18:    end for
19:  end while
20: end function

```

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 $O(n)$. Similarly, the TD-Dijkstra-FSM algorithm solves the TDQPP-EM 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`.

5.4 Dijkstra-SL 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 TDQPP-EM. First, we run the classical Dijkstra to solve the TDQPP-EM 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 cost across each arc of the QPP-EM optimal solution considering time-varying speeds. Algorithm 6 computes driver costs considering the heaviest congestion ratio 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 3.

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

```

1: function DRIVER_COST_HEAVIEST( $\gamma_i^p(t)$ ,
   ( $i, j$ ),  $\mathcal{Z}, \mathcal{S}, \sigma_h$ )
2:    $h | \gamma_i^p(t) \in Z_h = [z_h, z_{h+1}[$ 
3:    $\tau_{ij}(\gamma_i^p(t)) \leftarrow (\sigma_h u_{ij}) / L_{ij}$ 
   return  $c_d \tau_{ij}(\gamma_i^p(t))$ 
4: end function
```

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

```

1: function FUEL_COST_OPTIMAL_SPEED( $\gamma_i^p(t)$ ,  $(i, j)$ ,  $\mathcal{Z}, \mathcal{S}, s^*$ )
2:    $h | \gamma_i^p(t) \in Z_h = [z_h, z_{h+1}[$ 
3:    $\tau_{ij}(\gamma_i^p(t)) \leftarrow L_{ij} / s^*$ 
4:    $\mathcal{F}_{ij}(\gamma_i^p(t)) \leftarrow \lambda(kN_e V + \varsigma \alpha(w + q)s^* + \varsigma \beta(s^*)^3) \tau_{ij}(\gamma_i^p(t))$ 
   return  $c_f \mathcal{F}_{ij}(\gamma_i^p(t))$ 
5: 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 TDQPP-EM 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 1 shows a portion of the geographical area. We have considered 56 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 [27]. Relevant speed observations were extracted, consolidated and stored to obtain historical congestion patterns in this network.

As shown in Table 2 we have designed 80 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 – 20 tons) and full truck load (TL – 25 tons).

Table 2: Test instances

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



Figure 1: 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. We observe that minimizing fuel consumption is equivalent to minimizing GHG emissions as one liter of diesel generates $0.00279 \text{ t } CO_2 \text{ e}$ [25]. 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-FSM, LB and UB. Then, the exact value of each solution is recalculated with Algorithms 1 and 3 according to FSM and CMEM to reflect the key elements of real road network considering time-varying speeds.

All algorithms are implemented in C++ 17 using JetBrains CLion C++ 2017 release 2 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.

6.3 Computational results and analysis

In this section we assess the effectiveness and robustness of the proposed algorithms. 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 con-

Table 3: Overview of experimental design

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

sumption 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. Globally, the average computation time of Dijkstra-SL, TD-Dijkstra-LTM and TD-Dijkstra-FSM is 0.15, 0.35 and 0.32 seconds, respectively. 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 made 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 the obtained solutions, we first observe that the TD-Dijkstra-FSM generates the best solutions for each optimization criteria. 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 criteria, we can see from Table 4 that over our 80 instances TD-Dijkstra-FSM produces an average travel time of 1,338.59 seconds, which is 0.9% lower than TD-Dijkstra-LTM (1,350.79 seconds) and 6.67% lower than Dijkstra-SL (1,434.25 seconds). This exposes the error margin associated with using the LTM or the speed limit instead of using exact calculations with the FMS. The fuel consumption reported by the TD-Dijkstra-FSM (under the travel time objective) is 9.9 *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 [31] in their annual statistical report.

When looking at the Fuel Consumption optimization criteria, the TD-Dijkstra-FSM algorithm minimizes the travel time and the fuel consumption, as expected. When Fuel Consumption is minimized instead of Travel Time, for TD-Dijkstra-FSM, the distance decreases, on average, by 10.69% from 20.82 to 18.60 *km* yielding a reduced amount of emissions of 5.68% for all instances. More specifically, for all D^* instances the distance decreases, on average, by 13.13% to create 7.80% savings in emissions. The distance decreases by 8.69% to create 3.98% savings in emissions for medium network instances M^* , and the distance decreases by 11.86% to create 4.36% savings in emissions for small network instances S^* . These results clearly show

Table 4: Algorithm performances under different optimization criteria

Optimization criteria	Instances	Dijkstra-SL				TD-Dijkstra-LTM				TD-Dijkstra-FSM						
		Dist	TT	Fuel	Cost	Sec	Dist	TT	Fuel	Cost	Sec	Dist	TT	Fuel	Cost	Sec
Travel Time	<i>L1-L20</i>	24310.27	1770.60	12.08	28.95	0.19	24845.29	1687.35	11.98	28.12	0.44	24901.75	1658.10	11.90	27.78	0.39
	<i>L1-M20</i>	10397.11	990.20	5.79	15.07	0.11	10886.42	875.90	5.57	13.85	0.26	10946.48	867.05	5.57	13.78	0.24
	<i>S1-S20</i>	6216.59	664.05	3.68	9.87	0.06	6910.06	569.85	3.57	8.95	0.15	6846.87	565.45	3.53	8.87	0.13
	<i>D1-D20</i>	42174.18	2312.15	19.28	41.83	0.24	39288.77	2270.05	18.13	40.14	0.56	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	22.76	0.35	20822.24	1338.59	9.90	22.76	0.32
Fuel Consumption	<i>L1-L20</i>	22321.74	2016.95	12.01	30.95	0.19	23072.33	1747.90	11.52	28.10	0.50	23094.44	1734.90	11.49	27.95	0.40
	<i>M1-M20</i>	9416.36	1110.45	5.80	16.11	0.11	9992.72	919.20	5.38	13.99	0.29	9994.84	910.95	5.35	13.90	0.23
	<i>S1-S20</i>	5701.79	715.25	3.64	10.26	0.06	6022.40	621.05	3.42	9.22	0.15	6034.97	605.80	3.38	9.03	0.12
	<i>D1-D20</i>	34468.35	2854.00	17.84	44.77	0.26	35092.44	2546.15	17.17	41.39	0.66	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	18597.36	1440.63	9.34	22.98	0.32
Cost	<i>L1-L20</i>	23208.80	1834.30	11.89	29.26	0.19	23970.88	1681.00	11.67	27.70	0.54	23935.03	1672.30	11.64	27.59	0.42
	<i>M1-M20</i>	9783.68	1052.75	5.79	15.60	0.11	10664.25	873.15	5.49	13.74	0.31	10669.63	870.20	5.48	13.70	0.24
	<i>S1-S20</i>	5784.21	702.65	3.64	10.15	0.06	6651.81	572.50	3.50	8.89	0.17	6668.82	568.75	3.49	8.85	0.13
	<i>D1-D20</i>	35977.58	2483.40	17.57	41.31	0.25	37154.89	2305.50	17.47	39.68	0.70	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	9.531	22.50	0.43	19549.86	1356.09	9.50	22.45	0.33

that for TD-Dijkstra-FMS, minimizing the travel time does not minimize the fuel consumption. In fact, even if the travel time increased globally by 7.08% (from 1338 to 1440 seconds), the consumption decreases 5.68% (from 9.89 to 9.34 *liters*). The same observation holds for TD-Dijkstra-LTM and Dijkstra-LS. We can conclude that when minimizing fuel consumption, our algorithms successfully manage the traffic congestion to find better paths.

The third part of Table 4 considers the cost-minimizing objective. Again, the TD-Dijkstra-FMS produces the best results with an average path cost of 22.45\$ compared to 22.50\$ and 24.08\$ with TD-Dijkstra-LTM and Dijkstra-SL. Minimizing the cost implies a compromise between the travel time (cost) of the drivers and the fuel cost. As the driver costs are the largest component of the total cost, the TD-Dijkstra-FMS solution obtained under the cost minimization criteria uses less travel time than when minimizing the fuel (1356.01 instead of 1440.63 seconds) but a little more fuel (9.50 instead of 9.34 *liters*). Therefore, the savings in travel time is up to 5.87% combined with a small increase in fuel consumption of up to 1.74%, leading to a reduction in the overall cost, on average, by 2.31% from 22.98 to 22.45 dollars. Further, we see from Table 4 that our TD-Dijkstra-FSM yields a global saving on GHG emissions and overall costs of 1.36% (22.45 versus 22.76 dollars) and 4.01% (9.50 versus 9.90 *liters*), respectively, between cost-minimizing and travel time minimizing paths.

We note that both TD-Dijkstra-LTM and TD-Dijkstra-FSM produce coherent results with respect to the optimization criterion used. Thus, when the travel time criteria is used, the TT is effectively lower with respect to its value under the other optimization criteria. This pattern is not respected by the Dijkstra-SL as the minimum cost (23.93) is obtained under the travel time optimization criterion. Finally, it is remarkable 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. Finally, Table 4 clearly shows that in the presence of traffic congestion, using a time-dependent algorithm (TD-Dijkstra-LTM or TD-Dijkstra-FMS) significantly enhances the quality of solutions with respect to a time-independent one (Dijkstra-SL).

The results shown in Table 5 further presents average results when the cost minimization objective is used for the Lower Bound, Upper Bound, Dijkstra-SL, TD-Dijkstra-LTM and TD-Dijkstra-FSM algorithms. It shows that both TD-Dijkstra algorithms consistently provide average solution costs bounded by the lower and upper bounds. Indeed, for the TD-Dijkstra-FSM the gap between the lower bound value and the cost minimizing paths is 16.3% (22.45 with respect to 19.30). However, the results of the time-independent Dijkstra-SL for L^* (29.26), M^* (15.60) and S^* (10.15) always exceed the corresponding upper bounds of 28.95, 15.07 and 9.87 dollars. These results clearly show that the reliability 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.

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 additional experiments conducted to assess the impact of traffic congestion on the travel time, fuel consumption and total cost. To this end,

Table 5: Average results under the total cost optimization criterion

Instances	Lower Bounds		Upper Bounds		Dijkstra-SL		TD-Dijkstra-LTM		TD-Dijkstra-FSM	
	Cost	Sec	Cost	Sec	Cost	Sec	Cost	Sec	Cost	Sec
L1-L20	22.74	0.28	28.95	0.19	29.26	0.19	27.70	0.54	27.59	0.42
M1-M20	10.43	0.16	15.07	0.11	15.60	0.11	13.74	0.31	13.74	0.31
S1-S20	6.62	0.09	9.87	0.06	10.15	0.06	8.89	0.17	8.85	0.13
D1-D20	37.41	0.37	41.83	0.25	41.31	0.25	39.68	0.70	39.67	0.55
Average	19.30	0.23	23.93	0.15	24.08	0.15	22.50	0.43	22.45	0.33

we now used the average results over the 60 instances L*, M* and S* with departure times fixed at 07h30, before the morning congestion, and 16h00 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.

If we look at the TD-Dijkstra-FSM with fuel consumption as the optimization criterion, we see that it increases from 12.75 to 13.64 *liters* (6.47%) when the path departure times are changed from 7h30 to 16h00. Similarly, the travel time increased, on average, by 11.55% induced by changes from 1,974.33 to 2,232.17 seconds. We observe the same pattern for the overall costs, which is increased, on average, by 9.25% induced by changes from 31.45 to 34.66 dollars. Overall, all algorithms produced expected results with respect to traffic conditions.

Table 6: Impacts of departure time

Algorithms	Optimization criterias	07h30				16h00			
		Avg Dist	Avg TT	Avg Fuel	Avg Cost	Avg Dist	Avg TT	Avg Fuel	Avg Cost
TD-Dijkstra-LTM	Travel Time	27416.54	1894.00	13.20	31.27	28898.15	2130.10	14.24	34.48
	Fuel consumption	25509.35	1988.60	12.78	31.61	26213.71	2247.63	13.67	34.82
	Cost	26949.01	1897.97	13.05	31.14	27507.23	2154.27	13.86	34.25
TD-Dijkstra-FSM	Travel Time	27395.37	1877.83	13.15	31.08	28911.75	2120.60	14.22	34.38
	Fuel consumption	25539.33	1974.33	12.75	31.45	26200.99	2232.17	13.64	34.66
	Cost	26851.93	1886.67	12.98	30.96	27422.03	2142.97	13.79	34.07

Figure 2 analyses in more details the impact of departure time on average path travel time and total cost for instances D1 to D20 (see Table 4). In Figure 2, the results of the TD-Dijkstra-FMS replicate the traffic pattern of Québec City with a moderate morning congestion between 7h30 and 9h00; low traffic between 10h00 and 14h30 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.

The final set of experiments presented in Table 7 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 the 60 L*, M* and S* instances. As expected, as the load increases, fuel consumption, and thus the cost, increase for both the TD-Dijkstra-LTM and the TD-Dijkstra-LTM. We can see that for both algorithms 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 when the load increase from 20 to 25 tons. For the TD-Dijkstra-FSM, the fuel

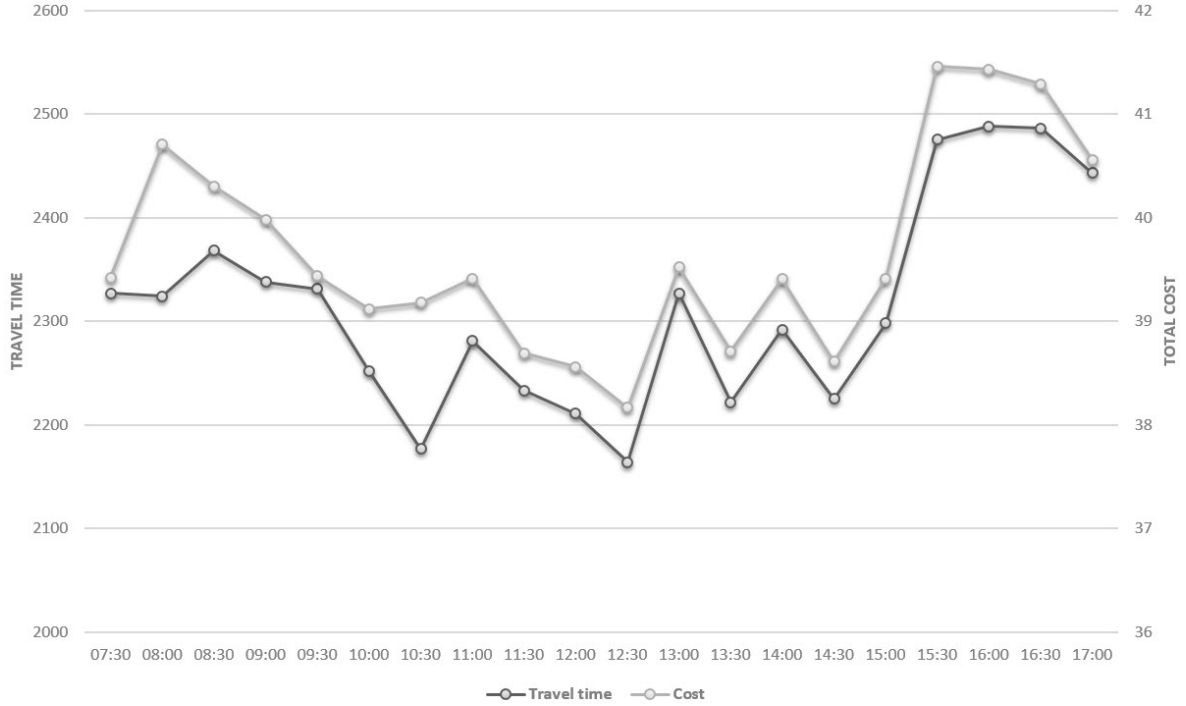


Figure 2: Impact of departure time

consumption increases from 11.45 to 12.59 *liters* (9.9%) from 15 to 20 tons and from 12.59 to 13.74 *liters* (9.1%) from 20 to 25 tons. This behavior is coherent with the fact that, proportionally, fuel consumption increases at a slowest rate with respect to the total load.

Table 7: Impact of carried load on performance measures

Algorithms	Performance measures	Carried Load		
		Empty Load (15 t)	LTL (20 t)	TL (25 t)
TD-Dijkstra-LTM	Avg Dist	27581.96	27524.62	27524.62
	Avg TT	2081.93	2084.43	2084.43
	Avg Fuel	11.46	12.61	13.77
	Avg Cost	30.88	32.22	33.55
TD-Dijkstra-FSM	Avg Dist	27661.85	27561.03	27515.65
	Avg TT	2072.1	2074.17	2074.17
	Avg Fuel	11.45	12.59	13.74
	Avg Cost	30.78	32.11	33.43

7 Conclusions

The TDQPP-EM 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. Time-dependent least cost lower

and upper bounds were derived based on QPP-EM properties. Fast and effective time-dependent Dijkstra label-setting algorithm and a lower bounding method have been implemented for eighty 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 algorithms are highly effective in finding good-quality solutions for benchmark instances of all size.

The extensive computational experiments demonstrated the benefit of choosing alternative paths in congested urban areas that leads to substantial fuel consumption/GHG emissions reduction and cost savings. We clearly demonstrate that using time-dependent algorithms lead to better results with respect to the ones which use constant speeds. Moreover, the required increase in computing time is negligible. 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 behavior with respect to optimization criteria, time of the day (level of congestion), carried loads and selected paths. We have also shown that carried loads affect the chosen path, particularly as the vehicle load becomes larger, the potential savings in fuel consumption and GHG emissions increase.

Further research should consider how to embed TDQPP-EM 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.

A An integer linear programming formulation for the TDQPP-EM

Let x_{ij} be a binary variable equal to 1 if and only if arc (i, j) is selected, and 0 otherwise, and y_{ijt} be a binary variable equal to 1 if and only if arc (i, j) appears in the solution at entering time t , and 0 otherwise. The formulation is given in a time-dependent network $\mathcal{G}_{\mathcal{T}} = (\mathcal{V}_{\mathcal{T}}, \mathcal{A}_{\mathcal{T}})$, which is expanded from the original (physical) time-dependent network $\mathcal{G}$ and time-varying travel time and cost. Specifically, $\mathcal{V}_{\mathcal{T}} = \{i_t | i \in \mathcal{V}, t \in \mathcal{T}\}$ represents the set of time-dependent nodes, in which each node $i_t \in \mathcal{V}_{\mathcal{T}}$ corresponding to a node-time pair indicates the state of node i and the arrival time t . The set of time-dependent arcs is denoted by $\mathcal{A}_{\mathcal{T}} = \{(i_t, j_{t'}) | (i, j) \in \mathcal{A}, t_0 \leq t \leq t_0 + H\delta\}$, in which time-dependent arc $(i_t, j_{t'})$ exists when the arc originates at a given physical node i at time t and terminates at a physical arc's terminal node j at the time t' . Thus, $t' = \gamma_j^p(t)$, such that $t_0 \leq t \leq t' \leq t_0 + H\delta$. For the same starting time in a time-expanded network, there are multiple time-dependent paths with different arrival times and travel costs. In order to define a standard flow balance constraint, a super-sink node $d'_{t_0+H\delta}$ is introduced to the

time-expanded network. It represents the end of the planning horizon, where all incoming arc travel times and costs have null values. For each physical node i , let its successors set be $\mathcal{E}^+(i) = \{j : (i, j) \in \mathcal{A}\}$. Similarly, $\mathcal{E}^-(i) = \{j : (j, i) \in \mathcal{A}\}$ denote the set of predecessors of node i . Furthermore, for each time-dependent node i_t , its successor node set is denoted by $\mathcal{E}_T^+(i_t) = \{j_{t'} : (i_t, j_{t'}) \in \mathcal{A}_T, t \leq t'\}$. Similarly, let $\mathcal{E}_T^-(i_t) = \{j_{t'} : (j_{t'}, i_t) \in \mathcal{A}_T, t' \leq t\}$ denote the set of predecessor nodes of the time-dependent node i_t . The following integer programming formulation of the TDQPP-EM is then proposed:

$$\min \sum_{(i,j) \in \mathcal{A}} \sum_{t \in \mathcal{T}} c_{ij}(t) y_{ijt} \quad (17)$$

subject to

$$\sum_{j \in \mathcal{E}^+(i)} x_{ij} - \sum_{j \in \mathcal{E}^-(i)} x_{ji} = \begin{cases} 1 & \text{if } i = o \\ -1 & \text{if } i = d \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

$$\sum_{j: (i,j) \in \mathcal{A}} x_{ij} \leq 1, i \in \mathcal{V} \quad (19)$$

$$\sum_{j: (i_t, j_{t'}) \in \mathcal{E}_T^+(i_t)} y_{ijt} - \sum_{j: (j_{t'}, i_t) \in \mathcal{E}_T^-(i_t)} y_{j_{t'}i_t} = \begin{cases} 1 & \text{if } i = o, t = t_0 \\ -1 & \text{if } i = d', t = t_0 + H\delta \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

$$\sum_{t \in \mathcal{T}} y_{ijt} = x_{ij}, (i, j) \in \mathcal{A} \quad (21)$$

$$x_{ij}, y_{ijt} \in \{0, 1\}, (i, j) \in \mathcal{A}, t \in \mathcal{T}. \quad (22)$$

The objective function (17) minimizes the overall travel cost in time-expanded network as in (14). Constraints (18) and (19) are defined to ensure the feasibility of an elementary path from origin o to destination d . Hence, constraints (18) correspond to the physical network flow balance. Constraints (19) guarantee that the selected physical arcs constitute a feasible path from the origin to the destination. Constraints (20) and (21) construct a corresponding time-dependent path in the time-expanded network based on the physical network. Additionally, space-time arc-to-link constraints (21) define the mapping between the physical and time-expanded networks. Finally constraints (22) define the domain and nature of the variables.

B Time-dependent lower and upper bounds for the TDQPP-EM

Proof 1 (Theorem 1). Given a solution path $p \in \mathcal{P}_\varphi$ it follows from (13) that:

$$\varphi(p_c, q_0, s^*) = \varphi(\Gamma(p_c, s^*)) + \varphi(\mathcal{F}(p_c, q_0, s^*)). \quad (23)$$

From (4) and (11) it results that the travel time can be expressed as (24) and fuel consumption and GHG emissions cost can be defined as (25), based on (8), (9) and (13):

$$\varphi(\Gamma(p_c, s^*)) = c_d \sum_{h=h_t}^{h_\gamma} \frac{l_{ij}^h}{s^*} = c_d \Gamma(p_c, s^*), \quad (24)$$

$$\varphi(\mathcal{F}(p_c, q, s^*)) = c_f (\lambda \varsigma \alpha w s^* + \lambda (k N_e V + \varsigma \beta (s^*)^3)) \sum_{h=h_t}^{h_\gamma} \frac{l_{ij}^h}{s^*} = c_f e_r(q, s^*) \Gamma(p_c, s^*), \quad (25)$$

where $e_r(q, s^*)$ is constant across all arcs of path p_c and represents the minimum fuel consumption rate. Combining (24) and (25) results in:

$$\varphi(p_c, q, s^*) = [c_d + c_f e_r(q, s^*)] \Gamma(\underline{\underline{p}}^*, s^*) = \varphi(\Gamma(\underline{\underline{p}}^*, s^*)). \quad (26)$$

As the first part of the equation (26) $c_d + c_f e_r(q, s^*)$ is constant then 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 the optimal solution for the TDQPP, i.e., $\underline{\underline{p}}^*$. Hence, an implication of (26) is that the optimal path $p_c^* = \underline{\underline{p}}^*$, which completes the proof of Theorem 1.

Proof 2 (Corollary 1.1). By observing that when the congestion ratios of all arcs are set to it lightest values $\sigma_h = \min_{ijh}(\sigma_{ijh})$ for each time interval Z_h it follows that the traversal time of a given path $p = (o = v_0, v_2, \dots, v_k = d)$ starting a time $t = 0$ is

$$\underline{\underline{\Gamma}}(p) = \sum_{l=1}^k \frac{L_{v_{l-1}v_l}}{u_{v_{l-1}v_l}} = \sum_{l=1}^k \int_t^{t+\tau_{v_{l-1}v_l}(t)} \sigma(\gamma) d\gamma = \int_{t_0}^{\underline{\underline{\Gamma}}(p)} \sigma(\gamma) d\gamma \quad (27)$$

where $\sigma(\gamma) = \sigma_h$. Furthermore, if we consider another path $p' \in \mathcal{P}_\Gamma$, from (27) it follows that:

$$\underline{\underline{\Gamma}}(p') \leq \underline{\underline{\Gamma}}(p) \Leftrightarrow \underline{\underline{\Gamma}}(p') \leq \underline{\underline{\Gamma}}(p), \quad (28)$$

which implies that:

$$\underline{\underline{\Gamma}}(\underline{\underline{p}}^*) = \underline{\underline{\Gamma}}(\underline{\underline{p}}^*). \quad (29)$$

As $\underline{\underline{p}}^* = \arg \min_{p \in \mathcal{P}_\Gamma} \{\underline{\underline{\Gamma}}(p)\}$ and $\underline{\underline{\Gamma}}(p_c^*) \leq \underline{\underline{\Gamma}}(p_c^*)$, from (29) it results that:

$$\underline{\underline{\Gamma}}(\underline{\underline{p}}^*) \leq \underline{\underline{\Gamma}}(p_c^*). \quad (30)$$

Hence,

$$\varphi(\underline{\underline{\Gamma}}(\underline{\underline{p}}^*)) \leq \varphi(\underline{\underline{\Gamma}}(p_c^*)). \quad (31)$$

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

$$\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*) \leq \mathcal{F}(p_c^*). \quad (32)$$

Then the following relationship also holds:

$$\varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*)) \leq \varphi(\mathcal{F}(p_c^*)). \quad (33)$$

Combining (32) and (33) yields:

$$\varphi(\Gamma(\underline{\underline{p}}^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*, q_0, s^*)) \leq \varphi(p_c^*), \quad (34)$$

which completes the proof of Corollary 1.1.

Proof 3 (Theorem 2). As p^* , $\underline{p}^*$, and $\underline{\underline{p}}^*$ are optimal solutions for the TDQPP, they are also feasible solutions for the TDQPP-EM. Additionally, $\underline{\underline{p}}_c^*$ is a feasible solution for the TDQPP-EM, then:

$$\varphi(p_c^*) \leq \varphi(p^*) \quad (35)$$

$$\varphi(p_c^*) \leq \varphi(\underline{p}^*) \quad (36)$$

$$\varphi(p_c^*) \leq \varphi(\underline{\underline{p}}^*) \quad (37)$$

$$\varphi(p_c^*) \leq \varphi(\underline{\underline{p}}_c^*). \quad (38)$$

By combining (35), (36), (37) and (38) we obtain the proof of the first part of the upper bound inequality:

$$\varphi(p_c^*) \leq \min\{\varphi(\underline{\underline{p}}_c^*), \varphi(p^*), \varphi(\underline{p}^*), \varphi(\underline{\underline{p}}^*)\} \leq \varphi(\underline{\underline{p}}^*) \quad (39)$$

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

$$\underline{\Gamma}(p) \leq \Gamma(p) \leq \frac{1}{\Delta}\underline{\Gamma}(p). \quad (40)$$

Combining (39) and (40) yields:

$$\Gamma(\underline{\underline{p}}^*) \leq \frac{1}{\Delta}\Gamma(p^*) \leq \frac{1}{\Delta}\Gamma(p_c^*), \quad (41)$$

which implies that:

$$\varphi(\Gamma(\underline{\underline{p}}^*)) \leq \frac{1}{\Delta}\varphi(\Gamma(p^*)) \leq \frac{1}{\Delta}\varphi(\Gamma(p_c^*)). \quad (42)$$

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

$$\varphi(\Gamma(\underline{\underline{p}}^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*)) \leq \frac{1}{\Delta}\varphi(\Gamma(p_c^*)) + \varphi(\mathcal{F}(\underline{\underline{p}}^*)). \quad (43)$$

An implication of (43) 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 2.

References

- [1] Barth, M. and Boriboonsomsin, K. (2008). Real-world carbon dioxide impacts of traffic congestion. *Transportation Research Record: Journal of the Transportation Research Board*, 2058:163–171.
- [2] Barth, M. and Boriboonsomsin, K. (2009). Energy and emissions impacts of a freeway-based dynamic eco-driving system. *Transportation Research Part D: Transport and Environment*, 14(6):400–410.
- [3] Bektaş, T. and Laporte, G. (2011). The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250.
- [4] Calogiuri, T., Ghiani, G., and Guerriero, E. (2015). The time-dependent quickest path problem: Properties and bounds. *Networks*, 66(2):112–117.
- [5] Cooke, K. L. and Halsey, E. (1966). The shortest route through a network with time-dependent internodal transit times. *Journal of Mathematical Analysis and Applications*, 14(3):493–498.
- [6] Dabia, S., Demir, E., and Van Woensel, T. (2017). An exact approach for a variant of the pollution-routing problem. *Transportation Science*, 51(2):607–628.
- [7] Dean, B. C. (2004). Algorithms for minimum-cost paths in time-dependent networks with waiting policies. *Networks*, 44(1):41–46.
- [8] Delling, D. and Nannicini, G. (2012). Core routing on dynamic time-dependent road networks. *INFORMS Journal on Computing*, 24(2):187–201.
- [9] Demir, E., Bektaş, T., and Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223(2):346–359.
- [10] Demir, E., Bektaş, T., and Laporte, G. (2014a). The bi-objective pollution-routing problem. *European Journal of Operational Research*, 232(3):464–478.
- [11] Demir, E., Bektaş, T., and Laporte, G. (2014b). A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793.
- [12] Di Bartolomeo, M., Grande, E., Nicosia, G., and Pacifici, A. (2017). Cheapest paths in dynamic networks. *Networks*, 55(2):23–32.
- [13] Ehmke, J. F., Campbell, A. M., and Thomas, B. W. (2016a). Data-driven approaches for emissions-minimized paths in urban areas. *Computers & Operations Research*, 67:34–47.
- [14] Ehmke, J. F., Campbell, A. M., and Thomas, B. W. (2016b). Vehicle routing to minimize time-dependent emissions in urban areas. *European Journal of Operational Research*, 251(2):478–494.

- [15] Fleischmann, B., Gietz, M., and Gnutzmann, S. (2004). Time-varying travel times in vehicle routing. *Transportation Science*, 38(2):160–173.
- [16] Franceschetti, A., Demir, E., Honhon, D., Van Woensel, T., Laporte, G., and Stobbe, M. (2017). A metaheuristic for the time-dependent pollution-routing problem. *European Journal of Operational Research*, 259(3):972–991.
- [17] Franceschetti, A., Honhon, D., Van Woensel, T., Bektas, T., and Laporte, G. (2013). The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological*, 56:265–293.
- [18] Ghiani, G. and Guerriero, E. (2014). A lower bound for the quickest path problem. *Computers & Operations Research*, 50:154–160.
- [19] Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107.
- [20] Huang, Y., Zhao, L., Van Woensel, T., and Gross, J.-P. (2017). Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological*, 95:169–195.
- [21] Ichoua, S., Gendreau, M., and Potvin, J. Y. (2003). Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research*, 144(2):379–396.
- [22] Jabali, O., Van Woensel, T., and de Kok, A.-G. (2012). Analysis of travel times and CO₂ emissions in time-dependent vehicle routing. *Production and Operations Management*, 21(6):1060–1074.
- [23] Kok, A. L., Hans, E., and Schutten, J. (2012). Vehicle routing under time-dependent travel times: the impact of congestion avoidance. *Computers & Operations Research*, 39(5):910–918.
- [24] Miller-Hooks, E. and Yang, B. (2005). Updating paths in time-varying networks given arc weight changes. *Transportation Science*, 39(4):451–464.
- [25] Ministère de l’Énergie et des Ressources Naturelles (2014). Facteurs d’émission et de conversion. Bureau de l’Efficacité et de l’Innovation Énergétique, Gouvernement du Québec. Available online (accessed on September 4, 2017).
- [26] Qian, J. and Eglese, R. (2016). Fuel emissions optimization in vehicle routing problems with time-varying speeds. *European Journal of Operational Research*, 248(3):840–848.
- [27] Renaud, J., Coelho, L. C., Belhassine, K., and Swadogo, M. D. N. (2017). Étude de réduction des émissions de GES chez Ameublements Tanguay par une optimisation des tournées de livraison en fonction du trafic routier (Rapport interne) . *Centre d’Innovation en Logistique et Chaîne d’Approvisionnement Durable, Université*

sité Laval / Rapport déposé au Bureau de l'efficacité et de l'innovation énergétique, Ministère des ressources naturelles, Gouvernement du Québec.

- [28] Savelsbergh, M. and Van Woensel, T. (2016). City logistics: Challenges and opportunities. *Transportation Science*, 50(2):579–590.
- [29] Sherali, H. D., Hobeika, A. G., and Kangwalklai, S. (2003). Time-dependent, label-constrained shortest path problems with applications. *Transportation Science*, 37(3):278–293.
- [30] Sung, K., Bell, M. G., Seong, M., and Park, S. (2000). Shortest paths in a network with time-dependent flow speeds. *European Journal of Operational Research*, 121(1):32–39.
- [31] Transports Canada (2017). Transportation in Canada 2016. Available online (accessed on September 4, 2017).
- [32] Van Woensel, T., Kerbache, L., Peremans, H., and Vandaele, N. (2008). Vehicle routing with dynamic travel times: A queueing approach. *European Journal of Operational Research*, 186(3):990–1007.
- [33] Wen, L., Çatay, B., and Eglese, R. (2014). 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.
- [34] Wen, L. and Eglese, R. (2015). Minimum cost VRP with time-dependent speed data and congestion charge. *Computers & Operations Research*, 56:41–50.
- [35] Yang, L. and Zhou, X. (2014). Constraint reformulation and a lagrangian relaxation-based solution algorithm for a least expected time path problem. *Transportation Research Part B: Methodological*, 59:22–44.