



HAL
open science

Green fleet management architecture: Application to economic itinerary planning

Guillaume Remy, Sara Mehar, Tonino Sophy, Sidi-Mohammed Senouci,
François Jan, Yvon Gourhant

► **To cite this version:**

Guillaume Remy, Sara Mehar, Tonino Sophy, Sidi-Mohammed Senouci, François Jan, et al.. Green fleet management architecture: Application to economic itinerary planning. 2012 IEEE Globecom Workshops, Dec 2012, Anaheim, United States. pp.369-373, 10.1109/GLOCOMW.2012.6477599 . hal-02557899

HAL Id: hal-02557899

<https://hal.science/hal-02557899>

Submitted on 19 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Green Fleet Management Architecture: Application to Economic Itinerary Planning

Guillaume Rémy^{*+}, Sara Mehar⁺, Tonino Sophy⁺, Sidi-Mohammed Senouci⁺, François Jan^{*}, Yvon Gourhant^{*}

^{*}Orange Labs, Lannion, France, ⁺DRIVE Laboratory, University of Bourgogne, France
guillaume.remy@ieee.org, {Sara.Mehar,Tonino.Sophy,Sidi-Mohammed.Senouci}@u-bourgogne.fr,
{yvon.gourhant,francois2.jan}@orange.com

Abstract—Over the last decade, great efforts are invested in reducing emissions of greenhouse gases at a world scale. We should therefore elaborate new global solutions to this type of pollution, in which fleets of vehicles are heavily contributing. In this paper, we propose a novel green architecture for global fleet management that considers the most important actors. The proposed architecture allows to collect useful information from several fleets of vehicles and their environment, and provides real-time services to companies that own one of these fleets. One important service is the calculation of economic itinerary. We propose a new "green" calculation method that is based on an efficient and high precision fuel consumption estimation. Our method uses: (i) topological information such as the elevation variations, source and destination of the trip, (ii) weather conditions, (iii) vehicle's characteristics, and (iv) driver's behavior. We implemented a lightweight traffic management server that provides a real-time service allowing to find the most economic itinerary for vehicles belonging to the fleet. The utilization of such architecture and mechanisms associated with itinerary planning results in a highly green fleet management system.

Index Terms—Traffic management, Green services and communications, Network architecture, Itinerary planning

I. INTRODUCTION

The climate of our planet is changing, and a lot of actions are being taken around the world to avoid reaching a no-return point that will compromise human life on earth. In order to slow down global warming, we are trying to reduce greenhouse gases emissions to their minimum. Fleets of vehicles are heavily contributing to these emissions. Most of the companies that own a fleet of vehicles are taking actions to reduce their effect (reduce the number of trips, replace the fleet with hybrid or electric vehicles, ...). Our solution aims at tackling the problem to a more global level: we propose to gather several fleets into one global fleet management architecture, providing global and more accurate services to the companies that own one of the fleets. It collects data from vehicles circulating on the road (mostly from vehicles of the fleets, but also from other vehicles that cross their ways) to calculate economic itineraries, with a high precision and efficiency. The system takes into account the collected data to estimate the location of traffic jams and special events. It also uses several APIs to calculate paths and retrieve data about elevation values and weather conditions.

The proposed architecture consists mainly of a main server, several company-specific servers, and vehicle-to-vehicle communications. The global server collects data from all the fleets,

and also caches data from APIs requests. Each company has an internal server that provides the itineraries planning platform to its fleet's users (the employees), and communicates with the main server to get global data and thus enhance the quality of the computed itinerary. When a fleet's vehicle navigates, it periodically checks with its company's server if the itinerary has to be updated. If there is an update pending, it accordingly updates the GPS device. The vehicle also stores information on-board, and sends the "debrief" information to the company's server when it arrives at destination. This communication can be done via WiFi if a Hotspot is available at the company's parking, or via LTE (Long Term Evolution) otherwise.

We implemented a lightweight company's internal server, that allows employees to book a car from the fleet and plan an economic itinerary. The itinerary calculation is based on the chosen car, the topology of the road (elevation variation, speed limits, ...), and weather conditions.

This paper is divided in four parts. In Section 2, we present related works on traffic and fleet management. Section 3 details the architecture we propose as a global fleet management system. Section 4 introduces a proof of concept that will be explained through a description of the implementation we made. Finally, we conclude about this work and our future research.

II. RELATED WORK

This section presents some existing works related to traffic and fleet management. The topic of green itinerary planning for a fleet of vehicles is not well covered and no publications have been found on this subject, and that is the reason why this particular topic will not be exposed in this section.

Fleet management is a topic that is now highly covered [1] [2] [3] [4] [5]. In [1], the authors implemented a web information system for transport telematic and fleet management. Their system is based on open-source softwares (MOWIS), and provides wide functionalities, such as vehicle tracking or fleet control and management. A transport control center collects all telemetric data from vehicles and stores them in a database. Users can manage the fleet on their mobile phone (using GPRS, WAP or SMS) or on any computer connected to the Internet, through a connection to the transport control center. The main advantage of this platform is that it is open-source: as well as our own solution that will be presented in

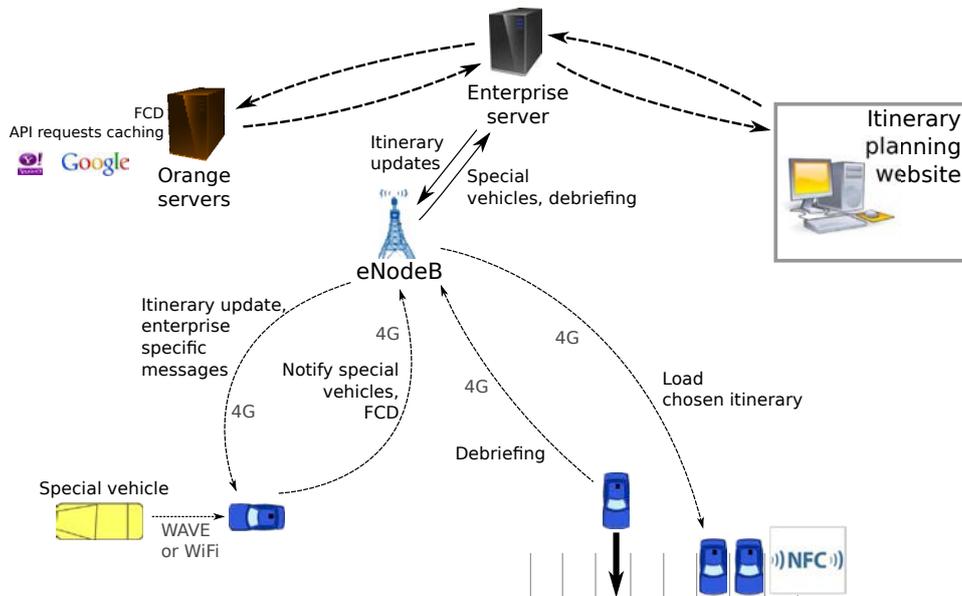


Figure 1. Global green fleet management architecture

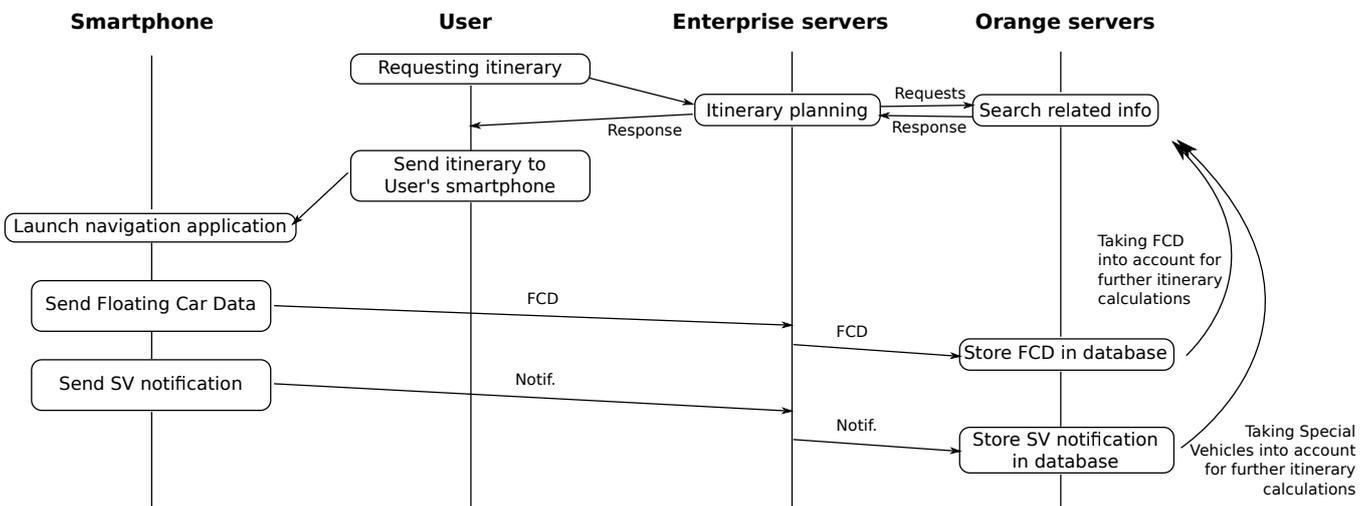


Figure 2. Sequence diagram

this paper, there is no proprietary locks, and new services can be freely, quickly and efficiently developed on the top of this platform. [5] presents some optimizations that can be made in car rental industry to better satisfy the customer and improve the efficiency of the fleet. Then, it proposes some powerful decision-making models, and more integrated notions of Fleet Assignment Problem (FAP) and revenue management.

A lot of traffic management systems have been proposed [6] [7] [8]. [6] proposes an evolutionary algorithm to avoid traffic congestions on the roads. Authors based their research on existing evolutionary algorithms for traffic management, and they exploited them on a six junction network, with traffic lights on each junction. They plan to develop an evolutionary technique that can perform a combined optimization of signal phasing offset and timing for the traffic lights. In [7], the

authors developed a fuzzy-based control algorithm that takes into account each vehicles safe and comfortable distance and speed adjustment for collision avoidance and better traffic flow. Their algorithm has proved to be very efficient, and was validated by an IEEE-802.11p-based communications study.

III. GLOBAL GREEN ARCHITECTURE

In this section we present the proposed architecture, and explain the role of each entity, as well as the communication technologies induced in the process. An important advantage of this architecture is its ability to be developed, implemented and deployed on already existing networks, and at a very low cost. Indeed, it only requires a main server (that can be placed anywhere on the Internet), a specific server for each company (i.e. for each fleet). It also requires an Android (or even iPhone) smartphone for each user, but as many people have a

smartphone nowadays, they can just install our application on their personal device.

As shown in Figure 1, the proposed architecture is constituted mainly of a global server (called Orange server), several company-specific servers (one of them is represented on the figure and called Enterprise server), and a vehicular network. The role of the main server is to gather data from all the fleets. This server can be a set of several physical servers, and it can be hosted by a company that will provide the fleet management services (Orange in our example). This main server collects Floating Car Data (FCD) and "special vehicle" notifications from the fleets' vehicles, and caches APIs data for further use. Communications between the different involved entities are represented as a sequence diagram in Figure 2.

Indeed, fleets' vehicles can detect a special vehicle and send a notification to the global server. The special vehicle (slow vehicle, road works vehicle, ...) notifies itself by broadcasting periodic messages to its neighborhood via 802.11p. These messages include information about the itinerary of the special vehicle, so that other vehicles can plan a route that will avoid the special vehicle. Fleets' vehicles can receive these messages when they are close to the special vehicle. Then, they send a notification via LTE to the global server (see Figure 3). The global server takes the notification into account for further itinerary calculations, and, if necessary, triggers an itinerary update for one or several fleets' vehicles. This "special vehicle" information can be used by any company-specific server, to take the special vehicle's itinerary into account when calculating an itinerary for a fleet's vehicle.

	Vehicle's ID	Latitude	Longitude	Itinerary size	Itinerary payload
Bytes	2	4	4	2	min 16

Figure 3. "Special vehicle" message

The remaining of this section describes the steps involved in the whole process of itinerary planning. When an employee needs a vehicle from the fleet of its company, he books the vehicle on a dedicated company's website. This website allows the user to book the vehicle, but also to plan his trip. When planning the trip, the website sends a request to the main server (Orange server on Figure 1). The main server takes the responsibility of providing the most economic itinerary to the company's server (called Enterprise server on Figure 1). It will use an API to compute all the available routes for the given source and destination (Google Maps in our example), and determines speed limits for each road segment. Then, it uses an API to retrieve the elevation of multiple points on the paths of the different routes. The server also uses another API to fetch weather conditions on multiple points on the routes paths (Yahoo! Weather in our example). Finally, it searches the previously collected FCD to know if there is any traffic jam on the planned path. All this process leads to a very high precision fuel consumption estimation: we take into consideration the speed limit of each road segment, the weather conditions (wind speed, rain, snow, ...), road elevation variations, real-time road

traffic, and of course the characteristics of the vehicle (type of engine, weight, mechanical characteristics, type of fuel, ...). This calculation will be more detailed in the next section.

After the successful calculation of the itinerary, the user validates his choice. The route can be either directly uploaded to the GPS of the chosen vehicle, or uploaded to the smartphone of the user, which can then serve as a navigation system in the vehicle. In this latter case, a suggested deployment is to equip each fleet's vehicle with a Near Field Communication (NFC) tag on the support that will hold the smartphone, providing a way to automatically trigger the navigation on the selected route when clipping the smartphone on its support. Our implemented Android application also automatically pushes a notification on the smartphone when a user validates an itinerary he planned on the website.

Once the vehicle is actually traveling on the road, it periodically stores information in a black box, and checks with Enterprise server if there is any update to its route. It also sends a "special vehicle" notification when it meets one.

When the vehicle arrives at its destination, it can send the "debrief" information (an aggregation and compression of the data stored in the black box) to the Enterprise server either via LTE, or via a WiFi access point located near the parking place (we can suppose that a Hotspot is deployed near each fleet's parking).

IV. PROOF OF CONCEPT

This section describes how we implemented a part of the previously presented architecture. Indeed, to prove the feasibility and efficiency of the proposed architecture, we are planning to test it on a real fleet of several vehicles. Other fleets of vehicles will be emulated, due to evident resources limitations.

The existing implementation consists of the most important and complex part of the system: the main and company-specific servers. We implemented a high precision fuel consumption estimation, along with the website designed for the employees to book a fleet's vehicle (see Figure 4). All the API calls needed for the itinerary planning and fuel consumption estimation have been implemented, and the main server is able to cache results from these API calls to avoid any unneeded API call. The next paragraph details the three main types of API calls involved in our system.

Firstly, we need to make an API call to retrieve all the possible routes between the given source and destination. Indeed, so many companies are providing free services for calculating itineraries that it would be useless for us to develop a new complex algorithm to achieve similar results. After getting all potential itineraries, we need to calculate which one is the most economic (i.e. the less fuel-consuming). This is where our algorithm goes into action: it uses data resulting from two other types of API calls to calculate a fuel consumption estimation that takes into account: (i) topological information such as the elevation variations, source and destination of the trip, (ii) weather conditions, (iii) vehicle's characteristics (properties of the engine, RPM, ...), and (iv) driver's behavior.

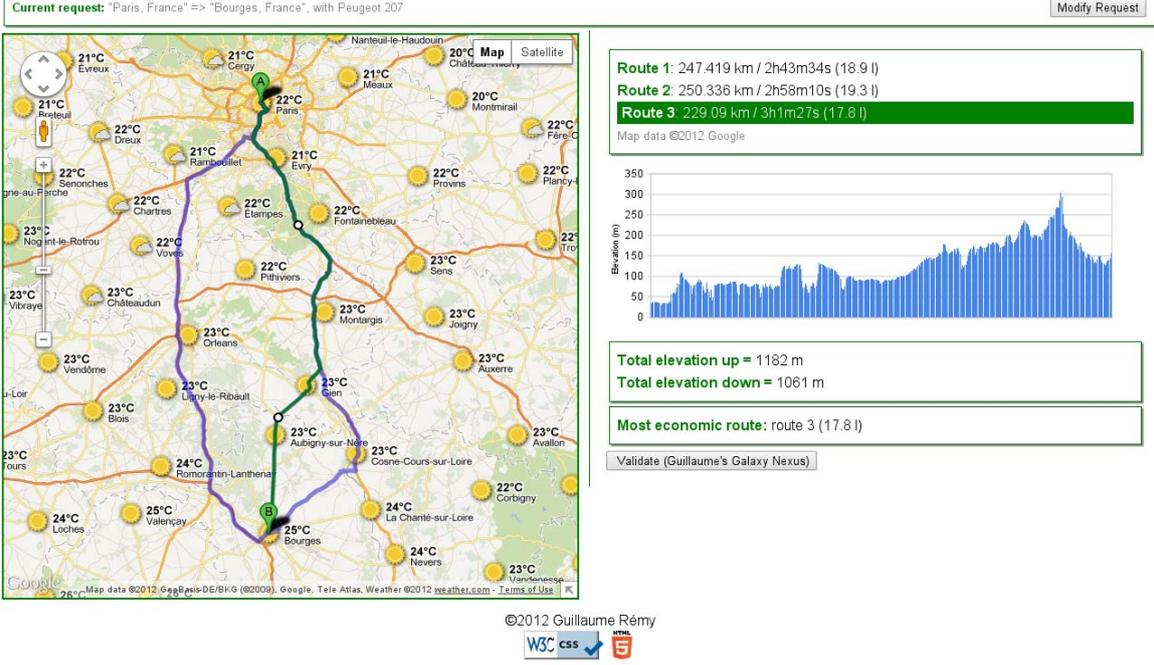


Figure 4. Itinerary planning website - Displaying results

Vehicles' characteristics and drivers' behaviors are stored in a database hosted by the Enterprise server. Of course, the privacy of such information is guaranteed by proper encryption and access management mechanisms.

The remaining of this section explains how we estimate the fuel consumption of a vehicle for a given itinerary at a given time. The equations listed below have been inspired by [9] and [10]. The result of the first API call (the one that fetches all available routes) allows us to retrieve multiple information such as length and speed limit for each segment (a segment is a part of the itinerary between two changes of direction). Then, we determine the points where we want to retrieve the elevation value and weather conditions. A point is placed systematically on each start of a segment. If a segment is longer than a given value, we divide it into fixed-length sub-segments, and a new point is placed at each end of these sub-segments. Then, we fetch the elevation value and weather conditions for each of these points. The last operation is the calculation of the fuel consumption for each sub-segment. This procedure is dynamic and is executed each time an itinerary is planned by a user. The results of these equations are obviously different for two distinct routes, but they can also be different the same route at two different points in time, because of the variation of weather conditions.

There is five steps involved in this calculation. First, we determine the force F needed by the vehicle to circulate at the given speed on the road. Twice, we deduce the power P from F . Third, we deduce the effective power P_e from

P . Fourth, we need to get the Specific Equivalent Fuel Consumption (SEFC) for this vehicle for the given Round-per-Minute (RPM) and equivalent pressure in the engine. Fifth, we deduce the actual fuel consumption, given $SEFC$ and P_e .

The first step is achieved by using Equation 1. m represents the vehicle's weight (in kilograms), i represents the inclination of the road, v represents the velocity of the vehicle (in meters per second), and $v_{relative}$ represents the relative velocity (difference between vehicle and wind speeds, in meters per second). $v_{relative}$ is calculated using the wind speed given by the weather conditions that were retrieved by the previous API call. We assume that the vehicle is always circulating at the speed limit of its road segment, except when there is a traffic jam (in that case the speed is reduced). i is given by another API call (the one that fetches the elevation values).

$$F = F_d + F_p + F_r \quad (1)$$

with $F_d = \frac{1}{2} \cdot \rho_{air} \cdot S_{C_x} \cdot v_{relative}^2$
and $F_p = m \cdot g \cdot \sin(i)$
and $F_r = m \cdot g \cdot 1.0246 \cdot 10^{-7} \cdot v^3$
 $- 1.3100 \cdot 10^{-6} \cdot v^2$
 $+ 1.7035 \cdot 10^{-5} \cdot v + 0.0116$

The second step is much easier: we just multiply the previously calculated F by the velocity of the vehicle (see Equation 2), and we obtain the power P .

$$P = F \cdot v \quad (2)$$

The third step leads us to the calculation of the effective power P_e . P_e is deducted from P and the ratio of the overall engine η (see Equation 3).

$$P_e = \frac{P}{\eta} \quad (3)$$

The fourth step involves the calculation of the Specific Equivalent Fuel Consumption (*SEFC*). *SEFC* is represented by a level curve given by the vehicle manufacturer. We performed an approximation of the actual curve, in order to be able to represent it as a function that can be interpreted by our program.

The fifth and final step gives the fuel consumption of the engine for the given road sub-segment. We calculate G_f (in gram per hour), and we multiply it by the duration of the trip on the sub-segment. Then, we have the weight of fuel that will be consumed for the sub-segment. Finally, we can multiply it by the density of the fuel to get a value in liters, which is much more useful for the user. G_f is obtained using the previously calculated *SEFC* and P_e (see Equation 4).

$$G_f = P_e \cdot SEFC \quad (4)$$

After we got all these values, we can sum all the G_f (see Equation 5, $G_f(i, j, k)$ being G_f for the k sub-segment of the segment j of the route i), then we can deduce which route is the most economic and provide the results to the user.

$$G_f^{\text{route } i} = \sum_{j=1}^{nb_{segments}(i)} \sum_{k=1}^{nb_{sub-segments}(i,j)} G_f(i, j, k) \quad (5)$$

V. CONCLUSION

In this paper, we propose a green fleet management system that uses collected data from fleets' vehicles and several free APIs to provide a reliable and economic itinerary. This system gathers several fleets, in order to increase the amount of collected data, and hence be more accurate in determining the position of traffic jams and special events. Moreover, it does not need any additional infrastructure on the road: it uses an embedded router in each vehicle and LTE, WiFi and WAVE (802.11p) communications. It is open-source and easily expandable with new services, and it is currently the only framework that provides such high precision fuel consumption estimations: elevation variations, weather conditions, advanced vehicle characteristics and driver's behavior are taken into account. Future work will involve completing the implementation of the overall architecture we presented in the paper, and real tests with many vehicles.

REFERENCES

- [1] S. Thong, C. T. Han, and T. Rahman, "Intelligent fleet management system with concurrent gps & gsm real-time positioning technology," in *Proc. ITST 2007, Sophia Antipolis, France*.
- [2] B. Predic, I. Antolovic, and S. Dordevic-Kajan, "Web information system for transport telematics and fleet management," in *Proc. TELSIKS 2009, Nis, Serbia*.
- [3] J. Grantner, B. Bazuin, L. Dong, J. Al-shawawreh, M. Castanier, and S. Hussain, "Condition based maintenance for light trucks," in *Proc. IEEE SMC 2010, Istanbul, Turkey*.
- [4] T. Duchrow, M. Schroer, B. Griesbach, S. Kasperski, F. Bermphol, S. Kramer, and F. Kirchner, "Towards electric mobility data mining," in *Proc. IEEE IEVC 2012, Greenville, South Carolina, USA*.
- [5] Y. Yang, W. Jin, and X. Hao, "Car rental logistics problem: A review of literature," in *Proc. IEEE SOLI 2008, Beijing, China*.
- [6] P. Balaji, G. Sachdeva, D. Srinivasan, and C.-K. Tham, "Multi-agent system based urban traffic management," in *Proc. IEEE CEC 2007*.
- [7] V. Milanes, J. Villagra, J. Godoy, J. Simo, J. Perez, and E. Onieva, "An intelligent v2i-based traffic management system," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 49–58, march 2012.
- [8] I. Leontiadis, G. Marfia, D. Mack, G. Pau, C. Mascolo, and M. Gerla, "On the effectiveness of an opportunistic traffic management system for vehicular networks," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1537–1548, dec. 2011.
- [9] J.-D. Wu and J.-C. Liu, "A forecasting system for car fuel consumption using a radial basis function neural network," *ELSEVIER Expert Systems with Applications*, vol. 39, 2011.
- [10] Z. Mohamed-Kassim and A. Filippone, "Fuel savings on a heavy vehicle via aerodynamic drag reduction," *ELSEVIER Transportation Research Part D*, vol. 15, 2010.