A General Constrained Optimization Framework for the Eco-Routing Problem: Comparison and Analysis of Solution Strategies for Hybrid Electric Vehicles

Giovanni de Nunzio, Ibtihel Ben Gharbia, Antonio Sciarretta

A General Constrained Optimization Framework for the Eco-Routing Problem: Comparison and Analysis of Solution Strategies for Hybrid Electric Vehicles

Giovanni De Nunzio[a], Ibtihel Ben Gharbia[a], Antonio Sciarretta[a]

[a]*IFP Energies nouvelles, 1 et 4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France.*

Abstract

Vehicles electrification marks a very important step towards sustainable mobility. However, energy efficiency and driving range of electrified vehicles are nowadays a major concern. From an algorithmic perspective, eco-routing opens up new possibilities regarding the strategies and tools aimed at improving energy efficiency by finding an energy-minimal route under different constraints coming from vehicle characteristics (powertrain, battery capacity, etc.) and user preferences (travel time, etc.). In this work, a powertrain-independent speed prediction model is presented. This model is then used to derive a fast numerical solution of the powertrain energy management for hybrid electric vehicles. Furthermore, a new general formulation is derived for the minimum-energy navigation problem, with a focus on the specific complexity introduced by electrified vehicles. The general constrained optimization problem is reformulated in several alternative ways in order to achieve a solution in limited computation time. The most commonly used approaches nowadays in the literature (integer programming and shortest path algorithms on directed graphs) are compared and benchmarked in terms of solution accuracy and computational effort. The objective is to identify best-practices in accurately and efficiently solving the constrained eco-routing problem.

Keywords: Eco-routing, constrained optimization, shortest path algorithm, integer programming, hybrid electric vehicles, energy management.

1. Introduction

Recently, e-mobility has been identified as an important means to reduce energy consumption and emissions of transportation. In an effort to comply with more and

Email addresses: giovanni.de-nunzio@ifpen.fr (Giovanni De Nunzio), ibtihel.ben-gharbia@ifpen.fr (Ibtihel Ben Gharbia), antonio.sciarretta@ifpen.fr (Antonio Sciarretta)

more stringent decarbonization measures, automakers are planning to significantly reduce the sales of vehicles solely powered by internal combustion engines (ICEs), and promote vehicles electrification. Market share projections seem to agree on the fact that purely electric vehicles (EVs) will reach at least 8% of all vehicle sales by 2025, while hybrid electric vehicles (HEVs) will rise to 23% of market share [1, 2]. Electric vehicles are battery-powered and the necessary electricity can be produced from regenerative sources. Furthermore EVs typically exhibit low emissions to their immediate environment in terms of combustion gases or noise levels. On the other hand, hybrid vehicles are equipped with both an internal combustion engine and an electric motor. The electric drive provides a better efficiency in many situations (e.g. for accelerating in city traffic) by allowing the combustion engine to operate at more efficient conditions. However, powertrain and technology evolution can only improve energy efficiency to a certain extent and require more and more to be complemented by on-board control and optimization strategies. Besides optimal energy management for HEVs and driving speed optimization (i.e. eco-driving) for improved efficiency, at route planning level eco-routing also appears to be a promising strategy. Eco-routing is the system that makes use of topological and traffic information about the road network to compute an optimal route in terms of energy consumption [3]. Such a strategy can be effectively used to reduce range anxiety and extend the driving range of EVs, but more in general it can be used to further reduce energy consumption for all types of vehicles. Attractive implementations exist in the literature for standard combustion-engine vehicles [4, 5, 6] and electric vehicles [7, 8], which have a single propulsion system. However, eco-routing for HEVs implies two major difficulties in the design both at modeling and routing level.

The first challenge consists in the accuracy of the energy consumption model, whose role is to estimate the energy cost on each elementary road segment. Energy cost of each segment (or link) of the road network depends on vehicle parameters and powertrain, but also on road slope and more importantly on the vehicle speed profile. While altitude can be obtained from geographical information services, speed profile prediction is particularly critical. Speed prediction methods appear to be well-established for HEVs predictive energy management applications on a given trip [9], but predicting a speed profile on each segment of a road network taking into account all the possible incoming and outgoing maneuvers is more challenging.

The second challenge in the eco-routing for HEVs consists in imposing constraints over the entire route in the optimization problem. Such constraints typically aim to enforce problem feasibility, by imposing that the battery state of charge (SOC) stay within the physical limits at any time along the route, or additional problem-related desired behaviors, such as a maximum travel time or a desired final SOC at the end of the trip. This problem is typically formulated either as an integer programming problem [10] or as a resource-constrained shortest-path problem (RCSPP) on graphs [11, 12], which are both known to be NP-hard problems.

The objective of this work is to give an overview of the alternative optimization for-

mulations for the constrained eco-routing problem that offer limited computation time and thus are more suitable for practical end-user implementation. To the best of our knowledge a thorough description and comparison of such practical problems and algorithmic solutions is lacking. The result of the presented analysis is aimed at identifying best-practices in the solution approach to the constrained eco-routing problem. The focus of the analysis is on HEVs because their powertrain is such that enough complexity is brought into the problem: the energy consumption modeling deals with an embedded optimal control problem (i.e. optimal energy management), and the routing optimization problem presents state dynamics (i.e. battery state-of-charge), state constraints and several decision variables. The problem formulations and the solution approaches in this work are applied to HEVs but could be transferred to other applications where state dynamics and state constraints appear in the optimization.

The contributions of this work are twofold. Firstly, within the scope of energy consumption modeling for HEVs, a fast numerical solution of the energy management system (EMS) for HEVs based on a deterministic speed profile prediction allows for a fast calculation of the fuel consumption per road segment as a function of the desired final SOC is proposed. Secondly and more importantly, a general problem formulation for the constrained eco-routing problem for HEVs is developed. Several alternative problem formulations for a fast solution of the constrained eco-routing problem are proposed, and search algorithms are also proposed to solve the presented optimization problems. The proposed approaches, making use of different techniques to deal with the constraints in a trade-off between solution accuracy and computation time, are compared in order to identify the best strategy to practically deal with such a complex problem. Discussions and experiment results comparing all the methods are presented.

The paper is organized as follows. Section II presents the predictive speed model, the general energy consumption model for all powertrains, and the simplified numerical solution method of the EMS for HEVs. Section III contains the general routing problem formulation for HEVs as a constrained optimization problem. Section IV describes the proposed alternative approaches for solving the problem and dealing with the SOC constraints, either via shortest path algorithms or via integer programming. Section V presents the search algorithms to solve the proposed formulations (pseudo-code of the algorithms is also provided in Appendix). The routing results and the comparison in terms of predicted fuel consumption and computation times are presented in Section VI.

1.1. Related works

Data-based driving behavior models are typically based on historical information about traffic conditions on the different portions of a road network. Speed and acceleration probability distributions and their statistical properties are generally used to represent driving behavior [13, 14] and to establish speed predictors. Those predictors either combine deterministic and stochastic approaches [15, 16], or are fully based on stochastic processes such as Markov chains [9, 17, 18], or are determined through

independent and identically distributed (i.i.d.) sequences [19]. These probability distributions are often obtained from standard driving cycles [20, 16] or real driving data [21, 17, 22, 19]. Although stochastic approaches may be more representative of the general driver behavior, they often fail to accurately represent acceleration/deceleration behaviors on small road segments and in presence of signalized intersections. Alternatively, link energy/emission costs may be directly estimated based on regression fits to empirical road data as in [23, 24, 25] without the need for a speed profile prediction. The two power sources of HEVs make fuel consumption estimation non-trivial, since it depends not only on the driving speed profile but also on the power split between the engine and the motor, which is dictated by the on-board EMS. Optimal EMS [26, 27] aimed at minimizing fuel consumption for a given battery consumption is obtained for a prescribed speed profile using the Pontryagin's minimum principle (PMP), which can be rather time consuming. In an attempt to reduce the computational load of the EMS, approximated optimization methods leverage the simple form of the predicted power demand at route planning level [28], or make use of historical information about average power split on recorded driving cycles [29, 30]. In this work, we extend the findings presented in [28] to define a fast numerical solution of the EMS for HEVs by predicting travel speed with a simple deterministic profile. Such a profile exhibits useful characteristics in terms of acceleration and power levels separation which allows the full optimal control problem of the EMS to be actually solved as a static optimization with a very limited number of decision variables.

In regard to the constrained routing optimization problem, some methods are proposed in the literature to solve such a problem in the case of eco-routing of HEVs. A fully-polynomial time approximation scheme [31], inspired by [32], was proposed to solve the constrained non-polynomial problem, with constraints on battery SOC feasibility. However, in the vehicle model, simultaneous use of the combustion engine and electric motor (i.e. hybrid mode) is not allowed. Also, the accuracy and the computational effort of the routing solutions is strongly dependent on the approximation parameter and the route length, showing scalability issues. Similarly, in [33] the complexity of the EMS is approximated by simple trade-off functions between the two power sources, and the routing algorithm with consideration of SOC constraints and recharging is an approximation inspired by [32]. The authors of [33] acknowledge that the proposed solution is still impractical in terms of computation time for an end-user deployment of the eco-routing strategy. They propose a workaround which consists in solving a shortest-path problem on a graph expanded with the battery SOC discretization, as a way to satisfy the SOC feasibility constraint by construction. The graph would then be such that only a choice among the feasible SOC options is allowed. Clearly, the accuracy and reliability of this approach are quite dependent on the chosen SOC discretization. Another interesting approach [34] solves the eco-routing problem for electrified vehicles by minimizing total trip time (including both the time on paths and at charging stations) with energy constraints, so that the vehicle is not allowed to run out of power before

reaching its destination. No constraint involving a desired final SOC is considered. The problem is simplified into a linear problem by decoupling route selection and recharging policy. Such a simplification is not possible for HEVs where the fuel consumption depends on the SOC variation and their optimal profiles cannot be sought separately. In [28], consideration of both SOC feasibility constraints and SOC terminal constraints leads to a constrained optimization problem. An iterative algorithm was proposed to solve the constrained shortest-path problem on a SOC-variation augmented graph. The graph expansion technique allowed the solution algorithm to find both the optimal route and the optimal battery discharge profile along the route. Graph expansions, as in [33], are in general well-known techniques to enforce routing constraints directly in the routing network. However, an analysis of which expansion technique is suitable for which application is lacking. In this work, we compare aforementioned and existing solution approaches, such as the battery expanded graph [33] and the SOC-variation expanded graph [28], with additional approaches, such as approximated constrained shortest-path algorithms or integer programming, especially formulated and proposed for a practical solution of the eco-routing problem for HEVs. The comparison is meant to shed light on the more effective methods to deal with such a complex problem.

2. Energy Consumption Model on Road Networks

The energy consumption prediction is paramount for a correct and reliable implementation of the eco-routing strategy. The goal of this section is to present a method to predict the energy consumption of various types of vehicles and powertrains in road networks. In the following, a physics-based approach will be used to estimate vehicle energy consumption, therefore a "synthetic" speed profile needs to be predicted on each road link and fed to the vehicle powertrain model.

2.1. Synthetic Speed Profile

The common assumption in routing applications is that the vehicle is subject to the general traffic conditions along its trajectory. However, at route planning level, exact speed information at any time and location is generally unavailable. Commercial mapping web-services generally provide aggregated traffic information in the form of an average speed $\bar{v}$. Typical aggregation intervals are of the order of the road segment and, temporally, of the order of minutes. The provided average speed $\bar{v}$ is constant on a road segment, and might slowly vary in time as the general traffic conditions evolve.

However, without any information on speed fluctuations around the average speed, the contribution of acceleration to energy consumption cannot be evaluated. That could lead to underestimating energy consumption especially in urban and/or suburban road networks. In fact, disruptions in the speed profiles and accelerations are more frequent at low velocity and caused not only by traffic, but also by road infrastructure. In particular, critical elements of the road infrastructure, such as traffic lights, intersections, and

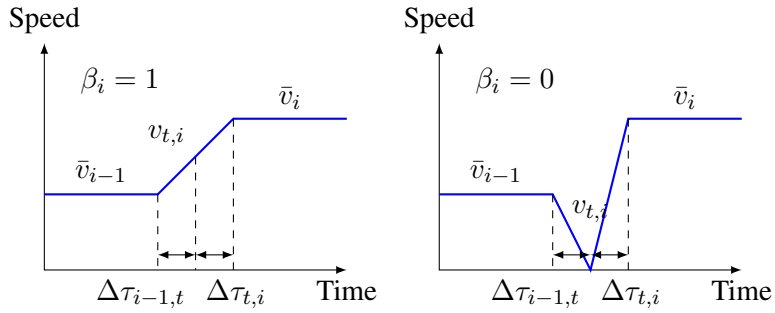

Figure 1: Interface accelerations: on the left-hand side a standard link transition, on the right-hand side a link interface with a stop sign.

turning movements are very likely to induce stops or significant deceleration [35]. In order to account for the effects of higher speed moments, such as infrastructure-induced acceleration and deceleration, and thus improve the energy consumption estimation, a synthetic speed profile may be generated as described in the following.

For each road segment i of the network, we assume that it is possible to know the segment length L_i, a prevailing average traffic speed $\bar{v}_i$, and the road grade α_i which varies within the considered segment depending on the position. The speed profile on segment i is supposed to be composed of two phases: a transition phase to go from $\bar{v}_{i-1}$, the cruising speed on the preceding segment, to $\bar{v}_i$, and a cruising phase at constant speed $\bar{v}_i$. Let us first introduce a transition speed at the interface between two segments defined as

$$v_{t,i} = \beta_i \frac{\bar{v}_i + \bar{v}_{i-1}}{2}. \tag{1}$$

where $\beta_i \in [0, 1]$ is a parameter depending on the type of interface (e.g. stop sign, traffic light, etc.), which could be selected in a deterministic or stochastic fashion.

The speed change between two road segments is modeled as two distinct transients: a first transient from $\bar{v}_{i-1}$ to $v_{t,i}$, and a second transient from $v_{t,i}$ to $\bar{v}_i$, both at constant acceleration/deceleration a_t (a model parameter), as shown in Figure 1. Let us recall that the sign function is defined as:

$$\text{sign}(z) = \begin{cases} +1, & \text{if } z > 0 \\ 0, & \text{if } z = 0 \\ -1, & \text{if } z < 0 \end{cases} \tag{2}$$

By considering time $\tau = 0$ at the beginning of the transition, the predicted speed on the road segment i can be thus written as:

$$v_i(\tau) = \begin{cases} \bar{v}_{i-1} + \text{sign}\left(v_{t,i} - \bar{v}_{i-1}\right) \cdot a_t \tau, & \tau \in [0, \Delta\tau_{i-1,t}] \\ v_{t,i} + \text{sign}\left(\bar{v}_i - v_{t,i}\right) \cdot a_t (\tau - \Delta\tau_{i-1,t}), & \tau \in (\Delta\tau_{i-1,t}, \Delta\tau_{t,i}] \\ \bar{v}_i, & \tau \in (\Delta\tau_{t,i}, \tau_i] \end{cases} \tag{3}$$

where the transient times are

$$\Delta\tau_{i-1,t} = \frac{|v_{t,i} - \bar{v}_{i-1}|}{a_t}, \quad \Delta\tau_{t,i} = \frac{|\bar{v}_i - v_{t,i}|}{a_t}. \tag{4}$$

Note that this synthetic speed model can depict behavior at traffic lights, stop signs and any other speed disruption caused by infrastructure or signalization, thanks to the choice of the parameter β. Also, it does not depict time spent idling at traffic lights or intersections, which is assumed not to contribute to energy consumption.

2.2. Vehicle Powertrain Model

After predicting the synthetic speed profile $v_i(t)$ on link i, the associated energy consumption can be evaluated using the vehicle longitudinal dynamical model hearby described, depending on the considered powertrain.

Considering the different on-board energy sources and propulsion architectures, modern road vehicles may be classified into three broad categories: internal combustion engine vehicles (ICEVs), electric vehicles (EVs), and hybrid-electric vehicles (HEVs). HEVs can be further classified into parallel, series, or series-parallel (or power-split, or combined) hybrids. All the latter types have one engine and one electric motor, most often powered by an electrochemical battery. In the rest of this work, parallel HEVs will be considered, for their property of having an electric machine mechanically coupled with the engine. HEVs have the advantage of combining the benefits of conventional combustion engines and electric motors. Depending on the application, the two propulsion systems may be configured to co-operate in order to achieve fuel consumption efficiency or increased power.

The vehicle longitudinal speed dynamics is typically expressed as:

$$m\frac{\mathrm{d}v(t)}{\mathrm{d}t} = F_w(t) - F_{\text{res}}(t) - F_b(t) - F_g(t), \tag{5}$$

where v is the vehicle speed, m is the vehicle mass, F_b is the mechanical brake force, F_g is the gravitational force, F_{res} is the sum of the resistance forces and is generally approximated by a polynomial function of v as

$$F_{\text{res}}(t) = a_2 v(t)^2 + a_1 v(t) + a_0, \tag{6}$$

with the coefficients a_0, a_1 and a_2 identified for a considered vehicle.

Hence, a general formulation of the powertrain force at the wheels, necessary to overcome the friction forces and allow the vehicle to move, would be as follows

$$F_w(t) = \frac{T_n(t)\gamma_t \eta_t^{\text{sign}(T_n(t))}}{r}, \tag{7}$$

where T_n is the torque generated by the propulsion system, r is the wheel radius, γ_t is the gear transmission ratio, and η_t is the transmission efficiency, which depends on the

gear ratio (although this dependency is often neglected). Therefore, independently of the type of powertrain, the available power at the wheels is defined as in [3]

$$P_w(t) = v(t)F_w(t), \tag{8}$$

2.2.1. Internal Combustion Engine Vehicles

In the case of ICEVs, the torque generated by the propulsion system corresponds to the combustion engine torque: $T_n(t) = T_e(t)$. The engine regime is related to the vehicle speed through the transmission ratio $\gamma_t = \gamma_e(t)$, and is defined as

$$\omega_e(t) = \frac{\gamma_e(t)}{r} v(t). \tag{9}$$

Assuming that a discrete transmission (gearbox) is used, γ_e varies with the gear selected either by the driver (manual transmission) or by the transmission controller (automatic transmission). The engine power is then defined as:

$$P_e(t) = T_e(t)\omega_e(t). \tag{10}$$

The fuel power consumed by the engine is usually modeled by means of steady-state tabulated experimental data (i.e. engine fuel map) as a function of engine rotational speed ω_e and engine torque T_e. For online applications, approximated closed-form expressions, such as polynomial models, are used. The Willans-line approach, for instance, consists in an affine representation relating the available engine power to the fuel power as [26]:

$$P_f(t) = a(\omega_e(t)) + b(\omega_e(t))P_e(t). \tag{11}$$

Empirical observations show that the two engine-speed dependent coefficients a and b can be further expressed analytically by means of polynomial functions of ω_e. Note also that the fuel power function has a discontinuity when the combustion engine is off (i.e. $P_e = 0$), because then $P_f = 0$.

Finally, the internal combustion engine fuel consumption can be expressed as

$$E_f = \int_0^{t_f} P_f(t)\,\mathrm{d}t, \tag{12}$$

where t_f is the duration of the considered time horizon.

2.2.2. Electric Vehicles

In the case of EVs, the torque generated by the propulsion system corresponds to the electric motor torque: $T_n(t) = T_m(t)$. The motor transmission ratio $\gamma_t = \gamma_m$ is considered fixed. The motor regime and the vehicle speed are then related by

$$\omega_m(t) = \frac{\gamma_m}{r} v(t). \tag{13}$$

The electric power supplied to or generated by the motor is then defined as:

$$P_m(t) = T_m(t)\omega_m(t), \tag{14}$$

and it is lower bounded by $P_{m,\min}(\omega_m)$ to limit the amount of energy per unit time that could actually recovered via regenerative braking.

The electrochemical power drained from or supplied to the battery is usually related to the electric motor power either by motor maps or by approximating polynomial functions, particularly suitable for online use:

$$P_b(t) = c(\omega_m(t)) + d(\omega_m(t))P_m(t) + e(\omega_m(t))P_m(t)^2. \tag{15}$$

The coefficients c, d and e are also time-varying and depend on the instantaneous values of the motor rotational speed along the driving profile.

Finally, the electric motor energy consumption is defined as

$$E_b = \int_0^{t_f} P_b(t)\,\mathrm{d}t. \tag{16}$$

2.2.3. Hybrid Electric Vehicles

Finally, for HEVs, the torque generated by the propulsion system corresponds to the sum of the electric motor and the combustion engine torque: $T_n(t) = T_m(t) + T_e(t)$. The total power demand at the hybrid propulsion system is then defined as:

$$P_d(t) = P_e(t) + P_m(t). \tag{17}$$

Instead of an overall energy consumption, one is more often interested in evaluating the minimal fuel consumption E_f for a given electric consumption E_b or SOC variation. The main degree of freedom to achieve this goal is represented by the split of power delivered by the engine and the motor. Such an optimal control problem (OCP) is usually formulated by considering the instantaneous cost function as a sum of the fuel consumption and an equivalent fuel consumption related to the battery SOC variation [26]. In the following, for clarity of notation, time dependency of the power is dropped, and the Hamiltonian function of the problem can be written as:

$$H(P_m, v(t)) = P_f\left(P_m, v(t)\right) + s \cdot P_b\left(P_m, v(t)\right). \tag{18}$$

The co-state adjoint to the SOC, s, is found such that the constraint over the desired final SOC is met.

A semi-analytical solution approach of the EMS optimization problem was presented in [35]. Thanks to the simple form of the synthetic speed profile defined in (3), it is assumed that the power demand profile can be divided into three distinct elementary power profiles corresponding to three different phases. In this work, by exploiting the choice of a constant acceleration a_t and such a simple form of the synthetic speed profile

$v_i(t)$, a further simplified formulation and numerical solution of the EMS is proposed. Let us define the power-split ratio $\mu = P_m/P_d$. Let us then assume that each period (or phase) j of the synthetic speed profile on link i is characterized by one value of the power split ratio μ_{ij}. Therefore, the following equations hold

$$P_{f,ij} = \begin{cases} a_{ij} + b_{ij}(1-\mu_{ij})P_{d,ij}, & \text{if } (1-\mu_{ij})P_{d,ij} > 0. \\ 0, & \text{else.} \end{cases} \tag{19}$$

$$P_{b,ij} = c_{ij} + d_{ij}\mu_{ij}P_{d,ij} + e_{ij}\mu_{ij}^2 P_{d,ij}^2. \tag{20}$$

The values μ_{ij} can be found with numerical methods as the solution of the following quadratically constrained linear program:

$$\begin{cases} \min\limits_{\mu_{ij}} & \sum\limits_{j=1}^{3} P_{f,ij}(\mu_{ij})\tau_{ij} & \text{(21a)} \\ \text{s.t.} & \sum\limits_{j=1}^{3} P_{b,ij}(\mu_{ij})\tau_{ij} = -\Delta_i \cdot C_b & \text{(21b)} \end{cases}$$

where τ_{ij} is the travel time on phase j of the synthetic speed profile of link i, Δ_i is the desired SOC variation to attain on link i, and C_b is the maximum energy stored in the battery.

Finally, the optimal fuel consumption of an HEV associated with the considered synthetic speed profile is

$$E_{f,i} = \sum_{j=1}^{3} P_{f,ij}(\mu_{ij})\tau_{ij}. \tag{22}$$

Intuitively, the optimal fuel consumption for an HEV is a function of the desired battery SOC variation Δ (i.e. $E_f(\Delta)$).

3. Routing Problem Formulation

The road transportation network can be conveniently modeled as a directed graph. Let $\mathcal{G} = (N, A)$ be such a graph, where N is the set of road intersections (or nodes), and A is the set of road arcs i (or links) connecting the nodes of the graph. However, the vehicle speed model presented in Section 2.1 depicts the interface accelerations between adjacent links in order to increase the speed prediction accuracy, and this is incompatible with such a standard representation of the road network. In particular, every node of the graph with two or more incoming links is critical because the upstream speed $\bar{v}_{i-1}$ is not unique. Evidently, this prevents from assigning unique costs to the links of the graph.

Therefore, the road network is modeled as a directed *line graph*, which can be thought of as the graph of the allowed maneuvers. Its definition is given hereafter:

Definition 1. *The line graph $\mathcal{L}(\mathcal{G}) = (A, A^*)$ of a directed graph $\mathcal{G} = (N, A)$ has a node for each link in $\mathcal{G}$ and each link represents a pair of adjacent links in $\mathcal{G}$.*

Therefore, a weighting function for the links of the line graph $k \in A^*$ can be defined as $w^* : A^* \rightarrow W$. In standard navigation systems, the routing graph is weighted with link length or travel time. In eco-routing, the focus is on energy savings and the weight assigned to each link of the graph represents the associated energy consumption. Each weight W_k represents the consumption to perform the corresponding maneuver $k \in A^*$, and is defined as:

$$W_k = \begin{cases} E_{f,k}, & \text{for ICEVs} \\ E_{f,k}(\Delta_k), & \text{for HEVs} \\ E_{b,k}, & \text{for EVs} \end{cases} \tag{23}$$

where $E_{f,k}$ is the fuel consumption on link k, Δ_k is the SOC variation on link k, and $E_{b,k}$ is the electrical energy consumption on link k.

In the rest of this work, we will focus on the eco-routing problem formulation and solution approaches for hybrid electric vehicles, working in either charge-sustaining or charge-depleting mode. The application to this type of vehicles makes the problem more challenging because of the limited battery capacity, the battery state-of-charge constraints and the optimal trade-off between fuel and electrical energy consumption. Note that the EVs can be considered as a particular and simpler case of charge-depleting HEVs.

Within this navigation framework, a generic routing problem for electrified vehicles with battery state-of-charge dynamics and constraints can be written as an optimization on the line graph as follows:

Problem 1. *For a given directed line graph $\mathcal{L}(\mathcal{G}) = (A, A^*)$, find the succession of arcs $k \in A^*$, or path $\mathbf{p} \subset \mathcal{P}$, where $\mathcal{P}$ is the set of all paths in $\mathcal{L}(\mathcal{G})$, and the electrical energy consumption Δ_k on each arc k such that*

$$\begin{cases} \min\limits_{\zeta_k, \Delta_k} & \sum\limits_{k \in A^*} W_k(\Delta_k) \cdot \zeta_k & \text{(24a)} \\ \textit{s.t.} & \sum\limits_{k \in k^+(i)} \zeta_k - \sum\limits_{k \in k^-(i)} \zeta_k = \begin{cases} 1, & \textit{if } i = i^o \\ -1, & \textit{if } i = i^d \\ 0 & \textit{else} \end{cases}, \quad \forall i \in A & \text{(24b)} \\ & \sum\limits_{k \in k^+(i)} \zeta_k \leq 1, \quad \forall i \in A & \text{(24c)} \\ & S_i = \sum\limits_{k \in k^-(i)} \left(S_{i^-(k)} + f_k(W_k, \Delta_k) \right) \cdot \zeta_k \quad \forall i \in A & \text{(24d)} \\ & S_i \in [S_{\min}, S_{\max}] \quad \forall i \in A & \text{(24e)} \\ & \sum\limits_{k \in A^*} f_k(W_k, \Delta_k) \cdot \zeta_k = S_{i^d} - S_{i^o} \geq \Delta & \text{(24f)} \\ & \zeta_k \in \{0, 1\}, \, \Delta_k \in \mathbb{R} \quad \forall k \in A^* & \text{(24g)} \end{cases}$$

The objective function in (24a) represents the sum of the costs, or weights, assigned to each link k. The optimization aims to minimize such an objective function by selecting the optimal sequence of links via the binary decision variable ζ_k, as well as the optimal SOC variation on link k via Δ_k, now considered as a continuous decision variable. The many optimization constraints are interpreted as follows. Constraints (24b) enforce flow conservation, meaning that the decision variable ζ_k must be chosen in such a way that the difference between the number of arcs k exiting node i (i.e. $k \in k^+(i)$) and the number of arcs entering node i (i.e. $k \in k^-(i)$) is equal to 1 in the case of the origin node i^o, equal to -1 in the case of the destination node i^d, and equal to 0 otherwise. Constraints (24c) ensure that the outgoing degree of each node is at most one, basically enforcing that the optimal path is simple. The first-order dynamics (24d) specify that the battery state-of-charge S_i at each node i is determined by the state $S_{i^-(k)}$ at the upstream node $i^-(k)$ of each arc $k \in k^-(i)$ entering node i, and by the decision (or stage) cost $f_k(W_k, \Delta_k)$. The stage cost is defined as:

$$f_k(W_k, \Delta_k) = -\Delta_k \tag{25}$$

In HEVs, Δ_k represents the desired SOC variation on link k, which corresponds to the electricity consumption associated with the minimum fuel consumption resulting from the EMS optimization. Then, the local constraints (24e) impose that the state remains feasible at each node i and for any partial path, with $S_{\min}$ and $S_{\max}$ denoting the minimum and maximum allowed state-of-charge, respectively. Similarly, the terminal constraint (24f) imposes a final desired state at the destination node i^d. Depending on the type of electrified powertrain, the battery SOC at the end of the trip should match a prescribed value. For charge-sustaining HEVs, the final battery state-of-charge S_{i^d} should match the initial value S_{i^o}, thus $\Delta = 0$ in constraint (24f), where Δ denotes the desired SOC variation over the entire route. For charge-depleting HEVs, the battery may be depleted, thus $\Delta = -S_{i^o}$ in constraint (24f).

4. Solution Approaches

Problem 1 in its generic form is known in the literature as a mixed-integer dynamic optimization [36, 37]. Often, this kind of problems is solved via a preliminary transformation in a mixed-integer nonlinear program (MINLP), which results into a large nonlinear problem due to full discretization of the differential equations [37]. As a result, the MINLP strategy needs to handle problems that are often extremely large and difficult to solve.

In a real-world routing problem, the road network is usually very large, which makes Problem 1 practically intractable. In order to achieve practical solutions of the routing problem for electrified vehicles, and to ensure the scalability of the approach to larger road networks, Problem 1 needs to be adapted and reformulated. In particular, the state

dynamics in (24d), the local state constraints in (24e) and the continuous decision variable Δ_k need to be dealt with.

There exist mainly three methods to tackle the constrained eco-routing problem for HEVs, as formulated in Problem 1, in a practical manner and with limited computation time.

The first two methods discussed in this work consist in solving the routing problem on either the SOC-expanded graph (or SEG in this work) or the SOC-variation expanded graph (or DSEG in this work). Both approaches leverage the idea of discretizing and integrating the continuous decision variable Δ_k directly in the graph construction. They differ in the amount of problem constraints directly verified by the graph by construction, therefore the resulting graphs could be significantly different in size. The basic idea is that the problem is transformed into a more tractable one by using polynomial time (or pseudo-polynomial) search algorithms at the expense of increasing the memory occupancy of the routing graph. Given these two routing graph expansions, it is then possible to reformulate Problem 1 in equivalent ways and solve it either via (constrained or unconstrained) shortest path algorithms or via integer programming algorithms, such as the branch-and-bound algorithm.

The third method does not require any graph expansion because it is solely based on the integer programming framework, therefore the continuous decision variable Δ_k can be explicitly considered in the optimization. However, in order to make the problem more tractable, the local state constraints (24e) are neglected because state dynamics are not easily enforced in an integer programming framework (i.e. static optimization) unless the number of constraints is significantly increased, as mentioned before. Therefore, this third method, applied directly to the line graph as Problem 1, cannot be considered as an equivalent formulation because some constraints are not verified, but we will show that it offers a practical solution for charge-sustaining HEVs, for which it is safer to assume that the optimal SOC trajectory stays naturally away from the battery capacity bounds.

In this section, we will detail the three solution methods and present alternative problem formulations of Problem 1 adapted to be solved by shortest path algorithms or integer programming algorithms, discussing pros and cons of each. In general, shortest path algorithms have the advantage of being founded on the dynamic programming principle and satisfy the principle of optimality, thus making it more natural to deal with state dynamics and sequences of decisions. Also, they naturally verify flow constraints and simple path constraints, and their decision process is binary (link belonging to optimal path or not) as enforced by the binary decision variable constraint. On the other hand, the advantage of integer programming lies in the fact that it does not require a purely discrete framework, therefore it is possible to consider the continuous decision variable Δ_k explicitly in the optimization problem, thus avoiding the approximation error due to discretization. Also, integer programming offers a more versatile framework for direct consideration of constraints and solution algorithms are capable of relaxing the problem

as a linear program (LP), when possible, for reduced computational complexity.

4.1. ΔSOC expanded graph (DSEG)

The problem of finding the optimal decision variable Δ_k for each link of the graph while searching the optimal route from an origin to a destination has been addressed before in the literature [33]. In order to overcome the computational complexity of the constrained optimization, the state constraints can be relaxed by enforcing them directly in the graph construction and/or by adding penalization terms in the objective function, which in turn can be reduced to a single-objective via weighted-sum scalarization. Note that the decision variable Δ_k, representing the electrical energy use on each link of the road graph, although continuous, must be discretized in order to be depicted within the graph framework.

The DSEG method consists in integrating the additional decision variable Δ_k directly in the graph construction, so that the primary decision variable ζ can select both the links and the SOC variations in the new expanded graph. Therefore, the line graph $\mathcal{L}(\mathcal{G}) = (A, A^*)$ is augmented by creating as many copies of each arc $k \in A^*$ as the number of pre-defined discrete feasible values of SOC variation. Let us define the DSEG as $\mathcal{L}_\Delta(\mathcal{G}) = (A, A^*_\Delta)$ and a new weighting function $w^*_\Delta : A^*_\Delta \rightarrow W_\Delta$ for the links $l \in A^*_\Delta$ of the augmented graph. Each weight represents the optimal fuel consumption to perform the corresponding maneuver $k \in A^*$ for a given SOC variation Δ_l and is defined as:

$$W_{l,\Delta_l} = E_{f,l} \tag{26}$$

where $\Delta_l \in [\Delta_{k,\min}, \Delta_{k,\max}]$. All the link copies $l \in A^*_\Delta$ corresponding to the maneuver $k \in A^*$ share the same power demand profile $P_{d,k}$. The copies differ from one another in terms of fuel consumption because of the associated Δ_l. The SOC discretization step δ_b is a design parameter, chosen in a trade-off between accuracy and computational burden. Thus the number of copies, or SOC variation levels, for each arc $k \in A^*$ is given by

$$N_\Delta = \left\lceil \frac{\Delta_{k,\max} - \Delta_{k,\min}}{\delta_b} \right\rceil + 1. \tag{27}$$

The total number of links of the DSEG is then

$$|A^*_\Delta| = |A^*| \cdot N_\Delta. \tag{28}$$

Furthermore, the variation range of $\Delta_l \in [\Delta_{k,\min}, \Delta_{k,\max}]$ depends on the physical properties of the maneuver $k \in A^*$.

A specific analysis has been dedicated to the prediction of the SOC variation for a given trip [28]. The objective of this study was to provide an a-priori estimation of the feasibility envelope of SOC variation for each road segment, in order to limit the number of copies to only the physically attainable ones. A training data-set was created by generating 150 trips with randomly chosen parameters: number of road segments,

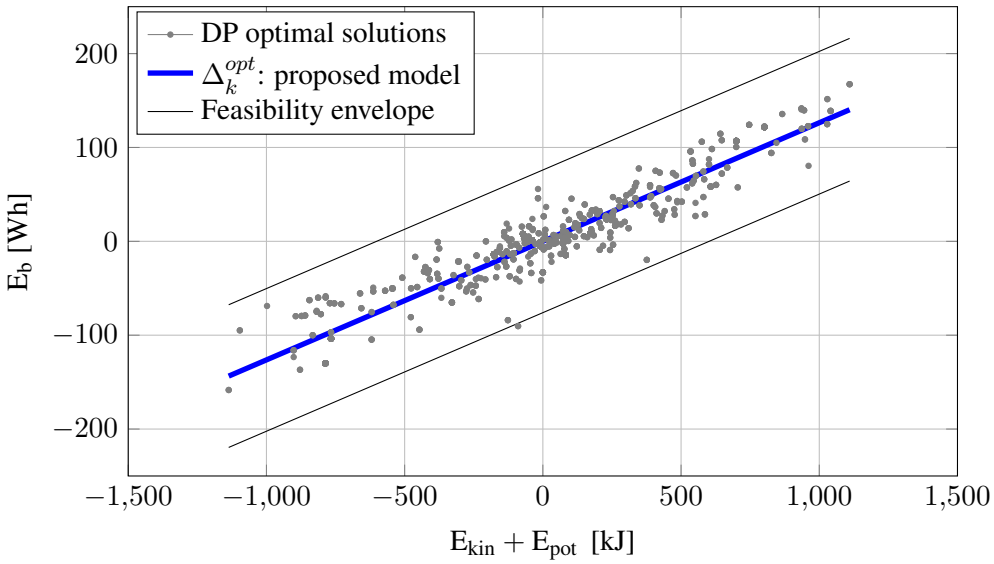

Figure 2: Validation of the proposed optimal SOC variation model against reference optimal results calculated via dynamic programming.

boundary speeds for each segment, average speed, mean road grade, and length of each segment. The OCP minimizing the fuel consumption over each generated trip was solved via dynamic programming (DP). The goal was to calculate the optimal Δ_k^{opt} for each arc $k \in A^*$ composing the trip, subject to the constraint of invariant total SOC (i.e. $\Delta = 0$). Inspired by the optimal results obtained via DP, a deterministic model for the optimal SOC variation per road segment has been proposed. This model states that the optimal SOC variation is a function of the predicted kinetic and potential energy to travel on the road segment, as well as of the vehicle parameters, and is defined as

$$\Delta_k^{opt} = \rho \frac{1/2\, m(v_{i,f}^2 - v_{i,0}^2) + mg(h_{i,f} - h_{i,0})}{C_b} \tag{29}$$

where $v_{i,f}$ and $v_{i,0}$ are the speed values at the end and the beginning of the road segment, $h_{i,f}$ and $h_{i,0}$ are the altitude values at the end and the beginning of the road segment. The correction parameter ρ is tuned in order to minimize the estimation error with respect to the optimal results obtained by the DP. Finally, in order to model the prediction uncertainty and have an estimate of the feasible SOC discretization range, the optimal values of Δ_k^{opt} obtained from the DP were used to compute confidence intervals. In Figure 2, the gray dots represent the optimal values of SOC variation Δ_k^{opt} on link k, and the solid blue line is the proposed optimal SOC variation model. The confidence intervals are computed in an intentionally conservative way by defining an envelope around the estimated values. Evidently, the conversion from the actual electrical energy consumption at the battery E_b to the SOC variation can be done based on the vehicle battery capacity.

In this work $C_b = 7.6\,\text{kWh}$, and therefore the feasible SOC variation $[\Delta_{k,\min}, \Delta_{k,\max}]$ is equal to $\pm 1\%$.

With the proposed graph expansion, the local constraint (24e) and the terminal constraint (24f) are not enforced directly in the graph, therefore the optimization problem would still be constrained. Problem 1 can be reformulated on the DSEG as follows:

Problem 2 (DSEG-P1). *For a given directed graph $\mathcal{L}_\Delta(\mathcal{G}) = (A, A^*_\Delta)$, find the succession of arcs $l \in A^*_\Delta$, or path $\mathbf{p} \subset \mathcal{P}$, where $\mathcal{P}$ is the set of all paths in $\mathcal{L}_\Delta(\mathcal{G})$, such that*

$$
\begin{cases}
\min\limits_{\zeta_l} & \sum\limits_{l \in A^*_\Delta} E_{f,l} \cdot \zeta_l & & & \text{(30a)} \\
\textit{s.t.} & \sum\limits_{l \in l^+(i)} \zeta_l - \sum\limits_{l \in l^-(i)} \zeta_l = \begin{cases} 1, & \textit{if } i = i^o \\ -1, & \textit{if } i = i^d \\ 0 & \textit{else} \end{cases}, & \forall i \in A & \text{(30b)} \\
& \sum\limits_{l \in l^+(i)} \zeta_l \leq 1, & \forall i \in A & \text{(30c)} \\
& S_i = \sum\limits_{l \in l^-(i)} \left(S_{i^-(l)} - \Delta_l \right) \cdot \zeta_l & \forall i \in A & \text{(30d)} \\
& S_i \in [S_{\min}, S_{\max}] & \forall i \in A & \text{(30e)} \\
& \sum\limits_{l \in A^*_\Delta} -\Delta_l \cdot \zeta_l \geq \Delta & & \text{(30f)} \\
& \zeta_l \in \{0, 1\}, & \forall l \in A^*_\Delta & \text{(30g)}
\end{cases}
$$

where $S_{i^-(l)}$ is the battery state-of-charge at the upstream node $i^-(l)$ of each arc $l \in l^-(i)$ entering node i.

Such a problem necessarily requires a constrained shortest-path algorithm to find the optimal solution, because, while looking for the path that minimizes the objective function, all other sub-paths need to be stored and verified against the state local and terminal constraints.

In order to overcome this issue and make the problem more tractable by using an unconstrained shortest-path algorithm, we propose a bi-level formulation in order to separate the search of the optimal path from the state constraints enforcement. In particular, an unconstrained shortest path algorithm can be used to solve the lower-level optimization, while a combination of iterative and binary search is used in the upper-level optimization to find the parameter (new decision variable) that verifies the constraints.

Problem 1 can be therefore alternatively formulated as a bi-level optimization as follows:

Problem 3 (DSEG-P2). *For a given directed graph $\mathcal{L}_\Delta(\mathcal{G}) = (A, A^*_\Delta)$, find the succession of arcs $l \in A^*_\Delta$, or path $\mathbf{p} \subset \mathcal{P}$, where $\mathcal{P}$ is the set of all paths in $\mathcal{L}_\Delta(\mathcal{G})$, and the parameter λ such that*

$$
\begin{cases}
\min\limits_{\lambda} & \left\| \left(\sum\limits_{l \in A^*_\Delta} -\Delta_l \cdot \zeta_l \right) - \Delta \right\| & \text{(31a)} \\
\textit{s.t.} & \zeta_l \in \arg\min\limits_{\zeta_l} \quad \sum\limits_{l \in A^*_\Delta} \left(\lambda \, E_{f,l} - (1-\lambda) \, \Delta_l \right) \cdot \zeta_l & \text{(31b)} \\
& \quad \textit{s.t.} \quad \sum\limits_{l \in l^+(i)} \zeta_l - \sum\limits_{l \in l^-(i)} \zeta_l = \begin{cases} 1, & \textit{if } i = i^o \\ -1, & \textit{if } i = i^d, \ \forall i \in A \\ 0 & \textit{else} \end{cases} & \text{(31c)} \\
& \quad \sum\limits_{l \in l^+(i)} \zeta_l \leq 1, \ \forall i \in A & \text{(31d)} \\
& \quad S_i = \sum\limits_{l \in l^-(i)} \left(S_{i^-(l)} - \Delta_l \right) \cdot \zeta_l, \ \forall i \in A & \text{(31e)} \\
& \quad S_i \in [S_{\min}, S_{\max}], \ \forall i \in A & \text{(31f)} \\
& \quad \zeta_l \in \{0,1\}, \quad \lambda \in [0,1] & \text{(31g)}
\end{cases}
$$

This new bi-level formulation is such that the problem aims to minimize fuel consumption and maximize battery energy recovery in the lower level, while enforcing the local and terminal SOC constraints in the upper level. More precisely, the lower level is formulated as a bi-objective optimization, where the parameter λ (i.e. the decision variable of the upper level) defines the trade-off between fuel and electrical energy consumption. Since λ is positive and bounded between 0 and 1, minimizing (31b) provides a sufficient condition for Pareto optimality [38]. In other words, all the obtained paths offer a Pareto-optimal trade-off between fuel consumption and battery charge. Furthermore, note that the second term added in the objective function (31b) can take on negative values. This implies that less efficient shortest-path algorithms are required to guarantee solution optimality (e.g. Bellman-Ford algorithm [39]).

The upper-level of the optimization problem is in charge of verifying the local and terminal state constraints. The terminal SOC constraint was relaxed and appears now in (31a), the objective function of the upper-level optimization. Therefore, the problem aims to minimize the difference between the actual total SOC variation and the desired one by choosing the decision variable λ (i.e. the optimization weight of the lower level).

In order to solve the upper level, a non-polynomial algorithm was proposed in [28] to find the parameter λ based on a combination of iterative and binary search. As mentioned, an unconstrained shortest-path algorithm (or even an integer programming algorithm) can be used in the lower-level, but one run of such an algorithm is performed at each search step of λ.

4.2. SOC expanded graph (SEG)

The SEG method consists in integrating both the additional decision variable Δ_k, similarly to the previous approach, and the local SOC constraint directly in the graph construction. Therefore, the graph expansion and the copies are performed for each node of the graph for predefined battery SOC levels in a discretized set.

Given the line graph $\mathcal{L}(\mathcal{G}) = (A, A^*)$ and the fully-disconnected graph $\mathcal{B} = (\mathcal{N}_b, \emptyset)$, with $\mathcal{N}_b$ representing the set of predefined levels of discretized battery SOC, the SEG is defined as the lexicographic product $\mathcal{L}_{SOC}(\mathcal{G}) = \mathcal{L}(\mathcal{G}) \cdot \mathcal{B} = (A_{SOC}, A^*_{SOC})$. For each node $i \in A$, we create as many copies $\kappa \in A_{SOC}$ as the pre-specified levels of battery charge.

In A^*_{SOC}, we create links for all possible connections between the nodes $\kappa \in A_{SOC}$. For each link $\xi \in A^*_{SOC}$ connecting two nodes copies $\kappa \in A_{SOC}$, we compute the associated SOC variation Δ_ξ and assign a weight equal to the corresponding fuel consumption. The dimension of the SEG is given by

$$|A_{SOC}| = |A|N_b, \qquad |A^*_{SOC}| = |A^*|N_b^2, \tag{32}$$

where $N_b = |\mathcal{N}_b|$ is the integer number of nodes copies (SOC discretization) and can be calculated as

$$N_b = \left\lceil \frac{S_{\max} - S_{\min}}{\delta_b} \right\rceil + 1, \tag{33}$$

where δ_b is the SOC discretization step as in the previous approach. A higher number of copies N_b, that is a smaller discretization step δ_b, corresponds to a finer precision.

The size of the SEG could be reduced with respect to (32) by considering only the links $\xi \in A^*_{SOC}$ corresponding to feasible SOC variations, as calculated for the DSEG. Hence, the dimension of the graph is reduced to

$$|A_{SOC}| = |A|N_b, \qquad |A^*_{SOC}| = |A^*| \cdot \left(N_b \cdot N_\Delta - \frac{N_\Delta^2 - 1}{4} \right), \tag{34}$$

where N_Δ is defined as in (27). Therefore, the weighting function for the links of the SEG $w^*_{SOC} : A^*_{SOC} \to W_{SOC}$ is such that each weight represents the optimal fuel consumption to perform the corresponding maneuver $k \in A^*$ for a given feasible SOC variation Δ_ξ and is defined as:

$$W_{\xi,\Delta_\xi} = E_{f,\xi}, \tag{35}$$

where $\Delta_\xi \in [\Delta_{k,\min}, \Delta_{k,\max}]$.

It is evident that the SEG size grows significantly, much more than the DSEG. Furthermore, the SEG is affected by the SOC discretization error, which is a function of the number of node copies N_b. This is due to the fact that the SOC discretization is imposed for the entire graph in the SEG method, while the link copies of the DSEG and the associated SOC variation levels are independent and can be different across the graph. Nevertheless, the advantage of the SEG is that the local constraint (24e) can be

directly encoded in the graph by creating copies in A_{SOC} in the range $[S_{\min}, S_{\max}]$, while the terminal constraint (24f) is easily verified by choosing the destination node corresponding to the desired final SOC.

Thanks to the fact that in the SEG the local constraint (24e) and the terminal constraint (24f) are enforced directly in the graph, the optimization problem can be formulated as follows:

Problem 4 (SEG-P). *For a given directed graph $\mathcal{L}_{SOC}(\mathcal{G}) = (A_{SOC}, A^*_{SOC})$, find the succession of arcs $\xi \in A^*_{SOC}$, or path $\mathbf{p} \subset \mathcal{P}$, where $\mathcal{P}$ is the set of all paths in $\mathcal{L}_{SOC}(\mathcal{G})$, such that*

$$\begin{cases} \min\limits_{\zeta_\xi} & \sum\limits_{\xi \in A^*_{SOC}} E_{f,\xi} \cdot \zeta_\xi & \text{(36a)} \\ \text{s.t.} & \sum\limits_{\xi \in \xi^+(\kappa)} \zeta_\xi - \sum\limits_{\xi \in \xi^-(\kappa)} \zeta_\xi = \begin{cases} 1, & \text{if } \kappa = \kappa^o \\ -1, & \text{if } \kappa = \kappa^d \\ 0 & \text{else} \end{cases}, \quad \forall \kappa \in A_{SOC} & \text{(36b)} \\ & \sum\limits_{\xi \in \xi^+(\kappa)} \zeta_\xi \leq 1, \quad \forall \kappa \in A_{SOC} & \text{(36c)} \\ & \zeta_\xi \in \{0,1\}, \quad \forall \xi \in A^*_{SOC} & \text{(36d)} \end{cases}$$

where κ^o is the origin node, and κ^d is the destination node in $\mathcal{L}_{SOC}(\mathcal{G})$. The arcs ξ exiting node κ are denoted $\xi \in \xi^+(\kappa)$, and the arcs entering node κ are denoted $\xi \in \xi^-(\kappa)$, analogously to the previous formulations. Note that, as already said, constraints (36b)-(36d) are naturally enforced by any shortest-path algorithm on graphs. Also, the objective function in (36a) can only take on positive values, since it represents fuel consumption. Therefore the problem can be solved by a single run of the Dijkstra algorithm in polynomial time. Integer programming algorithms may also be used here to solve the problem.

4.3. Relaxed problem on line graph (RLG)

The last approach presented here, as already mentioned, cannot be considered as an equivalent formulation of Problem 1 because it is assumed that constraints (24d) and (24e) can be neglected. Evidently, this is not generally true and there is no guarantee that the solution of the relaxed problem verifies such constraints. However, this relaxation makes the problem much easier to deal with and it could be considered as a practical approach especially for charge-sustaining HEVs. In fact, for this type of vehicles, it is generally observed that the optimal SOC trajectory hardly reaches the physical bounds of the battery and stays within a certain range around 50% of SOC. Furthermore, for this relaxed formulation, we will use the empirical observation that the fuel consumption of an HEV on each road segment, or arc, k is non-negative and varies linearly with the electrical energy use Δ_k. Such a linear relationship between fuel consumption and SOC variation for each arc can be inferred, for instance, via a linear regression on the discrete

SOC variation levels N_Δ, as defined in (27). The higher the number of such values, the higher the accuracy of the linear regression. Such a linear relationship for each arc can be calculated and stored offline, therefore the problem does not depend on the δ_b parameter.

Finally, the relaxed formulation can be written as follows:

Problem 5. *For a given directed line graph $\mathcal{L}(\mathcal{G}) = (A, A^*)$, find the succession of arcs $k \in A^*$, or path $\mathbf{p} \subset \mathcal{P}$, where $\mathcal{P}$ is the set of all paths in $\mathcal{L}(\mathcal{G})$, and the electrical energy consumption Δ_k on each arc k such that*

$$
\begin{cases}
\min\limits_{\zeta_k, \Delta_k} & \sum\limits_{k \in A^*} (\alpha_0 + \alpha_1 \, \Delta_k) \cdot \zeta_k & \text{(37a)} \\
\\
\textit{s.t.} & \sum\limits_{k \in k^+(i)} \zeta_k - \sum\limits_{k \in k^-(i)} \zeta_k = \begin{cases} 1, & \textit{if } i = i^o \\ -1, & \textit{if } i = i^d \\ 0 & \textit{else} \end{cases}, \quad \forall i \in A & \text{(37b)} \\
& \sum\limits_{k \in k^+(i)} \zeta_k \leq 1, \quad \forall i \in A & \text{(37c)} \\
& \alpha_0 + \alpha_1 \, \Delta_k \geq 0 & \text{(37d)} \\
& \sum\limits_{k \in A^*} -\Delta_k \cdot \zeta_k \geq \Delta & \text{(37e)} \\
& \zeta_k \in \{0, 1\}, \; \Delta_k \in \mathbb{R} \quad \forall k \in A^* & \text{(37f)}
\end{cases}
$$

where the fuel consumption of the HEV has been replaced in the objective function by a linear function of the decision variable Δ_k, as follows: $W_k(\Delta_k) = \alpha_0 + \alpha_1 \, \Delta_k$. The non-negativity of the fuel consumption is then imposed in the new constraint (37d). Note that this is still a MINLP, and in particular the objective function is bi-linear, presenting the product of the integer variable ζ_k and the linear variable Δ_k. However, this type of problems can be easily transformed into a MILP by introducing an additional auxiliary variable and by using the well-known convex relaxation with the McCormick envelopes [40]. Let us introduce then a new variable $w_k = \Delta_k \cdot \zeta_k$. The problem can be reformulated as follows:

Problem 6 (RLG-P). *For a given directed line graph $\mathcal{L}(\mathcal{G}) = (A, A^*)$, find the succession of arcs $k \in A^*$, or path $\mathbf{p} \subset \mathcal{P}$, where $\mathcal{P}$ is the set of all paths in $\mathcal{L}(\mathcal{G})$, the*

electrical energy consumption Δ_k and the variable w_k on each arc k such that

$$
\begin{cases}
\min\limits_{\zeta_k, \Delta_k, w_k} & \sum\limits_{k \in A^*} \alpha_0 \, \zeta_k + \alpha_1 \, w_k & \text{(38a)} \\
\textit{s.t.} & \sum\limits_{k \in k^+(i)} \zeta_k - \sum\limits_{k \in k^-(i)} \zeta_k = \begin{cases} 1, & \textit{if } i = i^o \\ -1, & \textit{if } i = i^d \\ 0 & \textit{else} \end{cases}, \quad \forall i \in A & \text{(38b)} \\
& \sum\limits_{k \in k^+(i)} \zeta_k \leq 1, \quad \forall i \in A & \text{(38c)} \\
& \alpha_0 + \alpha_1 \, \Delta_k \geq 0 & \text{(38d)} \\
& \sum\limits_{k \in A^*} -w_k \geq \Delta & \text{(38e)} \\
& w_k \geq \Delta_{k,\min} \, \zeta_k & \text{(38f)} \\
& w_k \geq \Delta_k + \Delta_{k,\max} \, \zeta_k - \Delta_{k,\max} & \text{(38g)} \\
& w_k \leq \Delta_k + \Delta_{k,\min} \, \zeta_k - \Delta_{k,\min} & \text{(38h)} \\
& w_k \leq \Delta_{k,\max} \, \zeta_k & \text{(38i)} \\
& \zeta_k \in \{0, 1\}, \; \Delta_k, w_k \in [\Delta_{k,\min}, \Delta_{k,\max}] \quad \forall k \in A^* & \text{(38j)}
\end{cases}
$$

This problem can be effectively solved via integer programming algorithms.

5. Search Algorithms

In the previous section, several alternative problem formulations of Problem 1 were given for a practical solution of the eco-routing problem for HEVs. An intuition about how the problem constraints are addressed by each formulation and what search algorithms can be used to effectively solve them was also provided. A summary of this discussion is given in Table 1, where it is shown at a glance what search algorithms can be used and compared to solve a same problem formulation, as well as the differences between the proposed formulations in terms of constraints enforcement.

In the remainder of this section, a more detailed description of the used search algorithms is provided. Some algorithms, such as the Dijkstra shortest-path algorithm, or the Bellman-Ford shortest path algorithm, or the integer programming algorithms are used in their standard implementation as provided by MATLAB R2019b or other well-known mathematical optimization solvers, such as Gurobi v9.0. These algorithms will not be detailed here, as sufficient documentation is publicly available online. All the other algorithms were adapted and tailored to the problem under consideration for improved performance.

5.1. Constrained Bellman-Ford shortest-path algorithm (CBF)

The computational complexity of the constrained BF algorithm is non-polynomial, typically exponential. This algorithm is generally impractical in most cases due to the

Algorithms \ Problems	DSEG-P1	DSEG-P2	SEG-P	RLG-P
Dijkstra (DIJ)			•	
Constrained Bellman-Ford (CBF)	•			
ε-approximated CBF (ε-CBF)	•			
Integer programming (IP)			•	•
Binary search + Bellman-Ford (BSBF)		•		
Binary search + IP (BSIP)		•		
Constraints \ Problems	**DSEG-P1**	**DSEG-P2**	**SEG-P**	**RLG-P**
local state constraints	✓	✓	✓	
terminal state constraint	✓	✓	✓	✓

Table 1: Summary of the search algorithms used to solve the proposed problems and overview of the type of constraints verified by each problem.

fact that all the sub-paths verifying the constraints to every node of the routing graph need to be stored during run-time. Implementations of such an algorithm exist in the literature [41, 42], but an adaptation to the case under analysis is provided here in Appendix in Algorithm 1. The proposed implementation still presents an exponential complexity, hence impractical for our purposes.

5.2. ε-approximated CBF (ε-CBF)

Promising polynomial-time approximation schemes for the constrained shortest-path algorithms have been proposed [32] by leveraging the idea of costs rounding and scaling and the idea of listing only Pareto-optimal sub-paths. This same idea has been applied to the search of optimal paths in graphs for EVs [33, 43] with SOC constraints and charging capabilities. The idea behind these approximation schemes consists in reducing the number of sub-paths that the constrained algorithm would store at each relaxation step by approximating the value of a cost (e.g. fuel consumption in this work) on an ε-spaced grid and storing the cost only if it is not dominated, in the sense of Pareto, by the previously stored ones [33]. The same approximation could be also applied to the value of a resource (e.g. electrical energy) to further reduce the number of retained sub-paths [43]. Evidently, such techniques give approximate solutions, and the level of approximation, as well as computational complexity, is determined by the choice of the sensitivity parameter ε. In this work, a modified version of such an algorithm is proposed in Algorithm 2. The main difference resides in the fact that the Pareto-optimality condition for the conservation of the sub-path is relaxed and all the sub-paths are stored. Note that, especially in the presence of local state constraints, a sub-path can

still be globally optimal and compliant with the constraints even though it is dominated in terms of fuel consumption or electrical energy consumption by another sub-path at an intermediate location. Furthermore, a modification to speed up convergence and reduce time complexity in the average case [44, 45] is applied here to the case of a constrained shortest path algorithm. The modification consists in introducing an early termination condition when an iteration of the algorithm main loop ends without making any sub-path update. When this happens, it means that the algorithm has already found all the shortest paths from the origin and any further iteration would not modify them.

5.3. Binary search

The proposed binary search algorithm to solve the bi-level Problem 3 is actually a combination of iterative and binary search. The objective of the iterative search is to find the two extreme values of the parameter λ which initialize the binary search. While the first initialization choice of $\lambda = 1$ for the higher bound of the binary search is feasible, making a second initialization choice of $\lambda = 0$ for the lower bound would result in solving a longest-path problem (NP-hard), which is both intractable and uninteresting for our needs. Therefore an iterative search of λ was proposed by means of a search step parameter. At the end of the iterative search, two extreme values of λ are available, with the corresponding paths verifying the local constraint (31f). The binary search was inspired by the one used in [35], but modified to include the verification of the SOC constraints and the termination condition to halt the recursion when a feasible solution is found. For more details, the pseudo-code is provided in Algorithms 3 and 4. Note that the optimal path search in Algorithm 4 could be performed by an integer programming algorithms instead of the Bellman-Ford algorithm. This is why a combination of binary search and integer programming (BSIP) is considered as a viable solution of problem DSEG-P2 in Table 1. However, its computational complexity makes it less appealing than the BSBF using an unconstrained shortest path algorithm in the lower optimization level.

6. Simulation Results

The simulation experiments were conducted with the objective of comparing the different approaches presented in Section 4 to solve the general Problem 1. The proposed strategies were implemented in MATLAB R2019b on a computer with CPU Intel(R) Core(TM) i7-8850H CPU at 2.6 GHz and 16 GB of RAM. For the following analysis, we considered three routing graphs of increasing size representing the road networks of the city center of Rueil-Malmaison (France), Aachen (Germany) and Paris (France). A visual representation of the three graphs is given in Figure 3, while the graphs size and how their size increases with the graph expansions described in Section 4 are indicated in Table 2.

The fuel consumption calculation (i.e. arc weights) for all the considered graphs was carried out by using the vehicle parameters of a common HEV and traffic conditions of

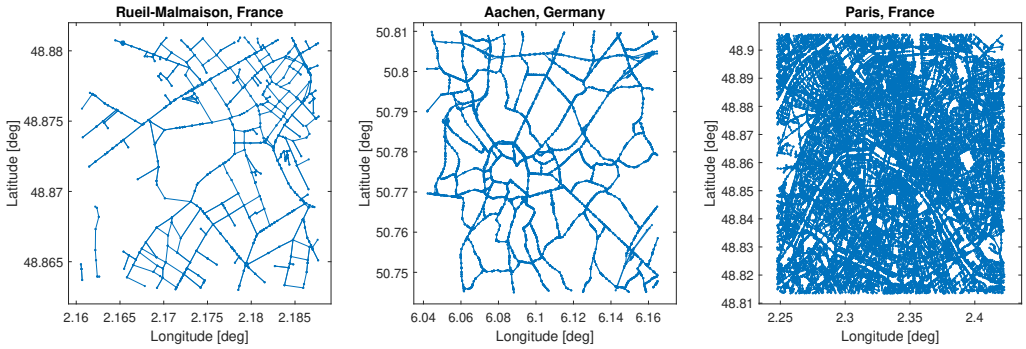

Figure 3: Considered routing graphs. From left to right, Rueil-Malmaison (France), Aachen (Germany) and Paris (France).

Graph	Rueil-Malmaison	Aachen	Paris
$\|V\|$	505	3,066	30,179
$\|A\|$	944	5,289	51,304
$\|A^*\|$	1,286	5,987	73,656
$\|A^*_\Delta\|$	11,574	53,883	662,904
$\|A^*_{SOC}\|$	2,300,654	10,710,743	131,770,584

Table 2: Graphs dimensions

regular working days at morning rush hour. The traffic data were retrieved from HERE Maps.

In the following, the different methods under analysis are compared firstly in terms of the trade-off between solution accuracy and computation time on a given routing graph, then their scalability is analyzed by testing their computation time on all considered graphs.

Let us recall the main parameters having an impact on the solution accuracy and computation time of each method. An overview of the parameter settings for the different problems and algorithms is given in Table 3.

It is possible to observe that the minimum and maximum battery SOC ($S_{\min}$ and $S_{\max}$, respectively) are used as parameters in all problems except for RLG-P, in which the local SOC constraints are not enforced. The minimum and maximum SOC variation per arc ($\Delta_{k,\min}$ and $\Delta_{k,\max}$, respectively) are used in all problems. A SOC variation range of $[-2, 2]\%$ was chosen, in order to be even more conservative than what suggested by the results in Figure 2. The SOC discretization step δ_b is also used in all problems except for RLG-P, which does not require any SOC discretization. The sensitivity to a variation of this parameter is studied in the following to assess its impact on solution accuracy and computation time. Some other algorithm-specific parameters were also varied in order to

Problems / Parameters	DSEG-P1	DSEG-P2	SEG-P		RLG-P
	ϵ-CBF	BSBF	DIJ	IP	IP
$S_{\max}$	100%				
$S_{\min}$	0%				
$\Delta_{k,\min}$	-2%				
$\Delta_{k,\max}$	2%				
δ_b	[0.1, 1]%				
ε_W	[10,30] Wh				
ε_Δ	[0.05,5]%				
δ_λ		0.1			
tol_Δ		[2,5]%			
γ_λ		1e-8			
γ_Δ		[0.1,10]%			

Table 3: Problems and algorithms parameter settings used in the experiments.

assess performance. These parameters are defined and used in Appendix in the pseudo-code of the algorithms, but for convenience let us recall their meaning. The algorithm ε-CBF makes use of the weight similarity parameter ε_W and the SOC similarity parameter ε_Δ to reduce the number of sub-paths stored in runtime. Finally, the bi-level optimisation problem DSEG-P2 makes use of the variation step δ_λ during the iterative initialization of the binary search; the tolerance tol_Δ on the difference between actual and desired final SOC is used to speed-up convergence and halt the optimization; the similarity parameters γ_λ and γ_Δ are used in the binary search to decide sensitivity to new solutions of the Pareto front. Note that the CBF and BSIP algorithms are voluntarily kept out of the analysis in the simulation experiments because of their excessively high computation time, which makes them uninteresting for our purposes of identifying the most effective solution strategies.

6.1. Trade-off between solution accuracy and computation time

In the first part of the experimental results, the different problems and algorithms are compared in terms of accuracy in the route fuel consumption estimation and computation time. To do so, the parameters shown in Table 3 are varied each within their specified range. The smallest routing graph among the considered ones (i.e. Rueil-Malmaison, France) was used in this experiment so that it was possible to test a wider range of parameters without incurring in extreme scalability issues. Also, for this experiment, since we are interested in the trade-off between accuracy and computation time, we limited the analysis to one O/D (origin/destination) pair, without loss of generality.

Furthermore, in order to properly compare the fuel consumption of routes with different final SOC values and evaluate the eco-route approximation error, a correction of the fuel consumption of each route for a same value of reference final SOC is neces-

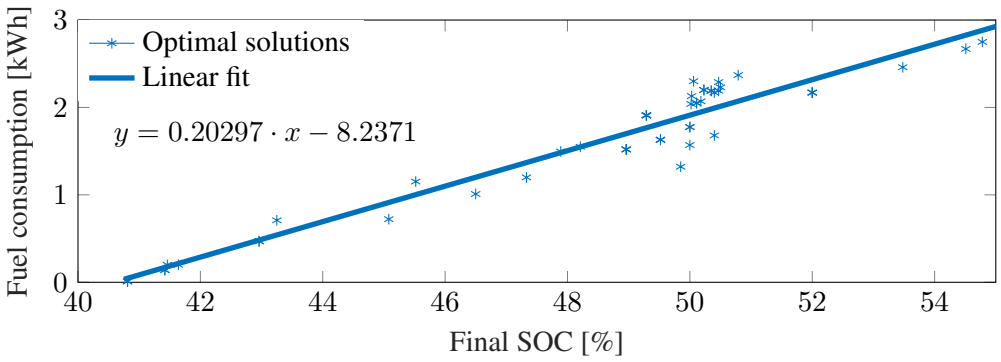

Figure 4: Linear fit of the relationship between fuel consumption and final battery SOC for a given O/D on the graph of Rueil-Malmaison ($S_0 = 50\%$). The data points represent the optimal route solutions obtained from all tested algorithms.

sary. To perform such a conversion, the different optimal solutions obtained from all tested algorithms for several combinations of parameters were represented in the fuel-consumption/final-SOC plane, as shown in Figure 4. Given the linear fit approximating the relationship between the optimal fuel consumption E_f and the final SOC S_f for the routes of the considered O/D pair, the corrected fuel consumption E_f^* for a same reference final SOC S_f^* can be calculated as follows:

$$E_f^* = E_f + 0.20297 \cdot \text{sign}\left(S_f^* - S_f\right) \cdot |S_f^* - S_f| \tag{39}$$

It is possible then to establish the sought trade-off between solution accuracy in terms of fuel consumption and computation time. The corrected fuel consumption corresponding to the reference final SOC $S_f^* = 50\%$ calculated with equation (39) is plotted as a function of the required CPU time for all tested algorithms in Figure 5. The algorithm ε-CBF on problem DSEG-P1 is able to find the route with the lowest corrected fuel consumption at the expense of a very high computation time. However, for a selection of parameters ensuring a practical computation time some of its solutions are dominated by other algorithms, thus becoming less appealing. Similarly, Dijkstra's algorithm on the SEG-P shows an improvement of fuel consumption accuracy with computation time but it remains dominated by the other methods for all selections of parameters. The algorithm BSBF on problem DSEG-P2 appears to have the most non-dominated points for low computation time, but with the disadvantage of not being able to further improve its fuel consumption accuracy because no selection of parameters would drastically increase its computation time. Finally, the IP algorithm on the RLG-P has just one point because it does not depend on any parameter. It offers an interesting trade-off in terms of fuel consumption and computation time for a small graph like the one considered in these tests, but it will be shown that it does not scale well with the graph size.

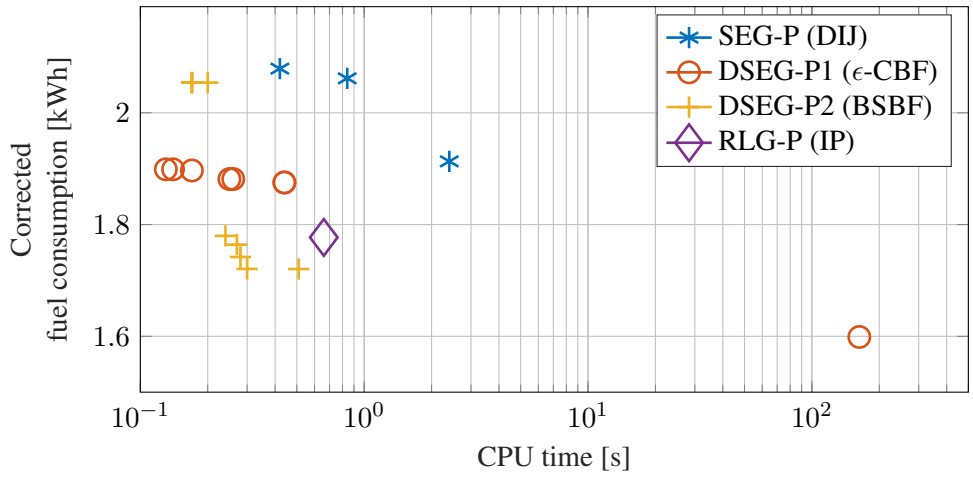

Figure 5: Trade-off between the optimal fuel consumption estimation accuracy and the computation time for the 4 tested algorithms and for different parameters selections. The indicated fuel consumption points correspond to a final SOC $S_f = 50\%$ for an initial SOC $S_0 = 50\%$.

Furthermore, in order to better understand the performance of the tested algorithms, it is important to discuss how the SOC discretization error impacts differently each problem. As already mentioned, the DSEG-P is based on a link-level graph expansion creating copies of the arcs weighted with fuel consumptions associated with different SOC variations. This property of creating only local copies of the links allows the approach to not have a fixed SOC discretization in the entire graph and to create copies corresponding to the actual calculated $\Delta_{k,j}$ independently of the values of the calculated SOC variations on other links of the graph. On the contrary, the SEG-P is mainly based on a node-level graph expansion, and the copies of the nodes must be created on a pre-defined SOC discretization grid determined by δ_b, in the same way in the entire graph. Often, in the literature, this difference between the pre-defined grid of desired SOC values at the nodes and the actual values calculated by the EMS is overlooked, and this may lead to a large inaccuracy in the fuel consumption estimation and the predicted SOC profile along the route.

To conclude this first set of simulation experiments, based on the results shown in Figure 5, we made the following selection of parameters for all algorithms in order to have the most accurate fuel consumption estimation with a computation time below 1 second (this seems reasonable for a graph the size of Rueil-Malmaison). Therefore, the SOC discretization step was set to $\delta_b = 0.5\%$; the ε-CBF parameters were set to $\varepsilon_W = 30\,\text{Wh}$ and $\varepsilon_\Delta = 5\%$; the BSBF parameters were set to $\text{tol}_\Delta = \gamma_\Delta = 1\%$.

For this choice of parameters and for the same given O/D pair on the graph of Rueil-Malmaison, an example of HEVs eco-routing is given in Figure 6. Interestingly, among the tested algorithms, only BSBF and IP calculate the same eco-route. The associated

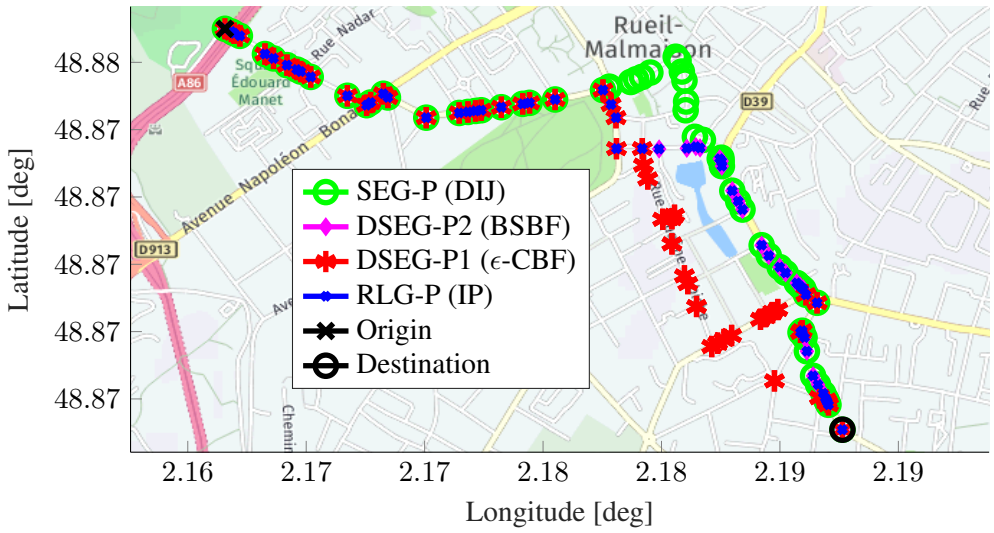

Figure 6: Map display of the calculated eco-routes on the graph of Rueil-Malmaison.

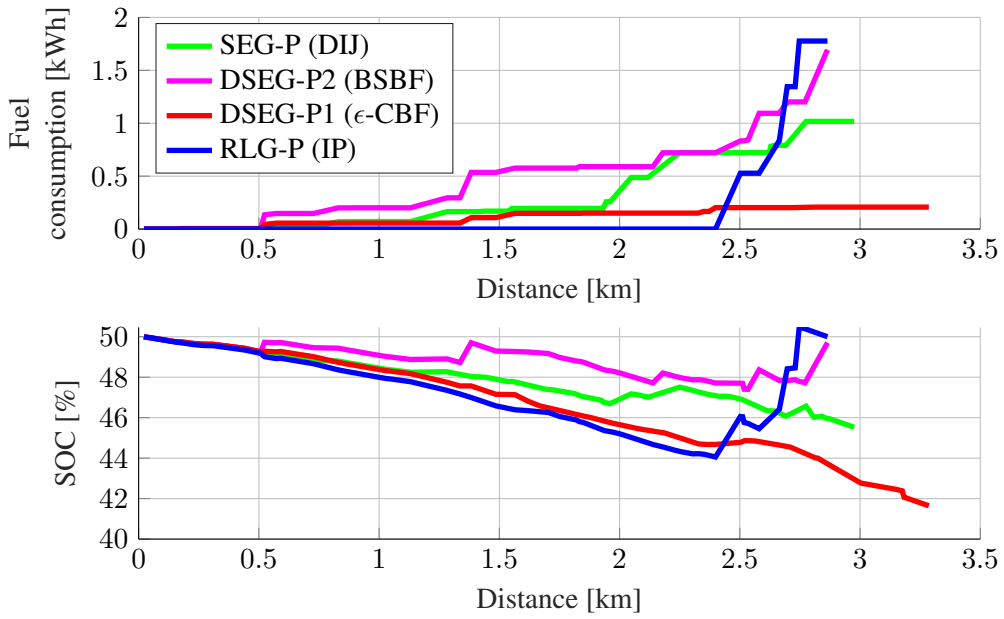

Figure 7: Fuel consumption and battery state-of-charge profiles of the eco-routes calculated on the graph of Rueil-Malmaison.

fuel consumption and battery SOC profiles along the routes are shown in Figure 7. Applying again the formula in (39), the BSBF algorithm yields the route with the minimum corrected fuel consumption for $S_f^* = 50\%$.

Problems / Graphs	DSEG-P1	DSEG-P2	SEG-P		RLG-P
	ϵ-CBF	BSBF	DIJ	IP	IP
Rueil-Malmaison	0.17	0.07	0.11	485.94	0.96
Aachen	42.59	0.17	0.87	–	3.69
Paris	–	4.06	17.38	–	366.88

Table 4: Average computation times (in seconds) of the tested algorithms for a random selection of 100 O/D pairs in each graph.

6.2. Computational performance and scalability

In this last set of experiments, after fixing the parameters for all algorithms as discussed before, we considered the three routing graphs and compared the computation times of the different considered methods on a random set of 100 different O/D pairs in each graph. We considered an initial SOC $S_0 = 50\%$. The routing graphs characteristics and the dimensions of their graph expansions are indicated in Table 2. In particular, we show that the SEG size grows significantly for large graphs.

The average computation times in seconds are stated in Table 4.

From this analysis it appears that, although the algorithms converge rather quickly to the optimal solution on the small graph of Rueil-Malmaison, they exhibit different scalability properties as the bigger graphs of Aachen and Paris are used. We also tested the integer programming algorithm on the SEG-P, but the already large size of the expanded graph of Rueil-Malmaison has a very negative impact on the computational performance of the algorithm, thus making it uninteresting with the SEG-P. As for the other algorithms, it is possible to observe that the most scalable one is the BSBF on the DSEG-P2, followed by the Dijkstra's algorithm on the SEG-P. However, the BSBF algorithm, as discussed in the previous set of experiments, exhibits higher accuracy in the fuel consumption estimation of the eco-route. Both the ε-CBF and the IP algorithm offer rather poor scalability properties and they are likely to be discarded when solving the constrained eco-routing problem on real-world routing graphs.

7. Conclusions

The objective of this work is to compare different practical (i.e. requiring limited computation time) solution approaches for the eco-routing problem of HEVs. The different methods are compared in terms of solution accuracy (i.e. eco-route fuel consumption estimation precision) and computation time, in order to identify the best procedures to tackle such a complex constrained optimization problem.

Firstly, this work presents methods to predict the energy consumption of various types of vehicles in road networks based solely on topological information coming from geographic information systems. The analysis then focuses mainly on hybrid electric vehicles, whose intrinsic constraints on battery capacity make the problem more complex.

A simplified energy management system for HEVs is proposed here for route planning applications, which allows for fast and accurate solutions, comparable to the optimal ones calculated offline via dynamic programming.

Then, several alternative formulations of the original eco-routing problem for HEVs are presented and discussed. The considered formulation based on different graph expansion techniques and the use of shortest-path algorithms, or based on problem relaxations and integer programming algorithms, offer suitable frameworks for a fast solution of the problem.

The algorithms that present the most appealing computation time for real-world eco-routing for HEVs appear to be the binary search and Bellman-Ford algorithm (BSBF) applied on the DSEG-P problem, followed by the Dijkstra's algorithm applied to the SEG-P problem, and finally by the integer programming applied to the RLG-P problem. The BSBF algorithm is the one exhibiting the best scalability properties and also the best algorithm in terms of fuel consumption estimation accuracy. To resume, the DSEG-P approach to the constrained eco-routing problem for HEVs in combination with the proposed BSBF algorithm is the most promising solution method. In alternative, the RLG-P and integer programming approach performs well for graphs of limited size. However, it should be reminded that some of the problem constraints (i.e. battery limits constraints) are relaxed in this approach to keep the computational burden at a low. This means that such method would not be suitable for solving the eco-routing problem for P-HEVs or even EVs, for which the battery could be fully depleted and constraints enforcing a minimum bound on the battery level would be required. On the other hand, the BSBF algorithm would present even better convergence performance when solving the eco-routing problem for charge-depleting electrified vehicles, thus confirming its versatility.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 724095 - ADVICE.

Appendix - Search algorithms pseudo-code

Algorithm 1 Constrained Bellman-Ford (CBF)

Input: $\mathcal{G} = (N, A)$ (routing graph), W (arc weights), Δ (arc SOC variations), u^o (source node), u^d (destination node), S_0 (initial SOC), S_f (desired final SOC), S_{min} (minimum allowed SOC), S_{max} (maximum allowed SOC)

Output: $\mathbf{p}$ (optimal path)

$\mathcal{P} \leftarrow \emptyset$ // list of all sub-paths
$\mathcal{D} \leftarrow \inf$ // list of optimal costs to every node
$\mathcal{S} \leftarrow \inf$ // list of optimal final SOC at every node
$\mathcal{P}(u^o) \leftarrow u^o$
$\mathcal{D}(u^o) \leftarrow 0$
$\mathcal{S}(u^o) \leftarrow S_0$
for $i \leftarrow 1$ to $(|N| - 1)$ **do**
  optimal $\leftarrow$ true
  for $j \leftarrow 1$ to $|A|$ **do**
    $u \leftarrow j^-$ // tail node of arc j
    $v \leftarrow j^+$ // head node of arc j
    for $k \leftarrow 1$ to $|\mathcal{P}(u)|$ **do**
      if $\left(\begin{array}{l} \mathcal{D}(u,k) + W(j) \notin \mathcal{D}(v) \text{ AND } \mathcal{S}(u,k) + \Delta(j) \leq S_{\text{max}} \\ \text{AND } \mathcal{S}(u,k) + \Delta(j) \geq S_{\text{min}} \end{array} \right)$ **then**
        $|\mathcal{P}(v)| \leftarrow |\mathcal{P}(v)| + 1$
        $\mathcal{P}(v) \cup [\mathcal{P}(u,k), v]$ // append v to the k^{th} path in the list of paths $\mathcal{P}(u)$
        $\mathcal{D}(v) \cup \mathcal{D}(u,k) + W(j)$
        $\mathcal{S}(v) \cup \mathcal{S}(u,k) + \Delta(j)$
        optimal $\leftarrow$ false
      end if
    end for
  end for
  if optimal **then**
    break // exit for loops
  end if
end for
$\mathbf{p} \leftarrow \mathcal{P}(u^d) \cap (\mathcal{S}(u^d) == S_f)$
return $\mathbf{p}$

Algorithm 2 ε-approximated CBF (ε-CBF)

Input: $\mathcal{G} = (N, A)$ (routing graph), W (arc weights), Δ (arc SOC variations), u^o (source node), u^d (destination node), S_0 (initial SOC), S_f (desired final SOC), S_{min} (minimum allowed SOC), S_{max} (maximum allowed SOC)

Output: $\mathbf{p}$ (optimal path)

Parameters: ε_W (weight similarity parameter), ε_Δ (SOC similarity parameter)

$\mathcal{P} \leftarrow \emptyset$ // list of all sub-paths
$\mathcal{D} \leftarrow \inf$ // list of optimal costs to every node
$\mathcal{S} \leftarrow \inf$ // list of optimal final SOC at every node
$\mathcal{P}(u^o) \leftarrow u^o$
$\mathcal{D}(u^o) \leftarrow 0$
$\mathcal{S}(u^o) \leftarrow S_0$
for $i \leftarrow 1$ to $(|N| - 1)$ **do**
  optimal $\leftarrow$ true
  for $j \leftarrow 1$ to $|A|$ **do**
    $u \leftarrow j^-$ // tail node of arc j
    $v \leftarrow j^+$ // head node of arc j
    for $k \leftarrow 1$ to $|\mathcal{P}(u)|$ **do**
      if $\left(\begin{array}{l} \mathcal{D}(u,k) + W(j) \notin \mathcal{D}(v) \text{ AND } \mathcal{S}(u,k) + \Delta(j) \leq S_{\text{max}} \\ \text{AND } \mathcal{S}(u,k) + \Delta(j) \geq S_{\text{min}} \text{ AND } |(\mathcal{S}(u,k) + \Delta(j)) - \mathcal{S}(v)| \geq \varepsilon_\Delta \\ \text{AND } |(\mathcal{D}(u,k) + W(j)) - \mathcal{D}(v)| \geq \varepsilon_W \end{array} \right)$ **then**
        $|\mathcal{P}(v)| \leftarrow |\mathcal{P}(v)| + 1$
        $\mathcal{P}(v) \cup [\mathcal{P}(u,k), v]$ // append v to the k^{th} path in the list of paths $\mathcal{P}(u)$
        $\mathcal{D}(v) \cup \mathcal{D}(u,k) + W(j)$
        $\mathcal{S}(v) \cup \mathcal{S}(u,k) + \Delta(j)$
        optimal $\leftarrow$ false
      end if
    end for
  end for
  if optimal **then**
    break // exit for loops
  end if
end for
$\mathbf{p} \leftarrow \mathcal{P}(u^d) \cap (\mathcal{S}(u^d) == S_f)$
return $\mathbf{p}$

Algorithm 3 Binary Search + BF

Input: $\mathcal{G} = (N, A)$ (routing graph), W (arc weights), Δ (arc SOC variations), u^o (source node), u^d (destination node), S_0 (initial SOC), S_f (desired final SOC)

Output: λ (optimization weight verifying the problem constraints)

Parameters: δ_λ (variation step of λ), tol_Δ (tolerance on the difference between actual and desired final SOC)

// Iterative initialization of binary search

$\lambda_1 \leftarrow 1$

$W^* = \lambda W - (1-\lambda)\Delta$

$\mathbf{p}(\lambda_1) \leftarrow \text{BF}\left(\mathcal{G}, W^*, u^o, u^d\right)$

$\lambda \leftarrow \lambda_1$

while $\Delta_{\mathbf{p}(\lambda)} < (S_f - S_0)$ **do** // $\Delta_{\mathbf{p}(\lambda)}$: total SOC variation on path $\mathbf{p}$ obtained for λ

  $\lambda \leftarrow \lambda - \delta_\lambda$

  $W^* = \lambda W - (1-\lambda)\Delta$

  $\mathbf{p}(\lambda) \leftarrow \text{BF}\left(\mathcal{G}, W^*, u^o, u^d\right)$

  if $\left|\Delta_{\mathbf{p}(\lambda)} - S_f + S_0\right| < \text{tol}_\Delta$ AND $\mathbf{p}(\lambda)$ verifies (24e) **then**

    return λ

  end if

end while

$\lambda_2 \leftarrow \lambda$

$\lambda \leftarrow \text{binarySearch}\left(\lambda_1, \lambda_2, \Delta_{\mathbf{p}(\lambda_1)}, \Delta_{\mathbf{p}(\lambda_2)}, \mathcal{G}, W, \Delta, u^o, u^d\right)$

return λ

Algorithm 4 binarySearch

Input: λ_1 (lower bound of the binary search), λ_2 (upper bound of the binary search), $\Delta_{\mathbf{p}(\lambda_1)}$ (total SOC variation on path $\mathbf{p}$ obtained for λ_1), $\Delta_{\mathbf{p}(\lambda_2)}$ (total SOC variation on path $\mathbf{p}$ obtained for λ_2), $\mathcal{G} = (N, A)$ (routing graph), W (arc weights), Δ (arc SOC variations), u^o (source node), u^d (destination node), S_0 (initial SOC), S_f (desired final SOC)

Output: λ

Parameters: γ_λ (λ similarity parameter), γ_Δ (SOC similarity parameter), tol_Δ (tolerance on the difference between actual and desired final SOC)

$\lambda \leftarrow (\lambda_1 + \lambda_2)/2$
$W^* = \lambda W - (1 - \lambda)\Delta$
$\mathbf{p} \leftarrow \text{BF}\left(\mathcal{G}, W^*, u^o, u^d\right)$
if $\left|\Delta_{\mathbf{p}(\lambda)} - S_f + S_0\right| < \text{tol}_\Delta$ AND $\mathbf{p}(\lambda)$ verifies (24e) **then**
  return λ
end if
if $\left(\begin{array}{l} |\lambda - \lambda_2| \geq \gamma_\lambda \text{ AND} \\ \left|\Delta_{\mathbf{p}} - \Delta_{\mathbf{p}(\lambda_2)}\right| > \gamma_\Delta \end{array}\right)$ **then**
  $\lambda \leftarrow \text{binarySearch}\left(\lambda, \lambda_2, \Delta_{\mathbf{p}(\lambda)}, \Delta_{\mathbf{p}(\lambda_2)}, \mathcal{G}, W, \Delta, u^o, u^d\right)$
end if
if $\left(\begin{array}{l} |\lambda - \lambda_1| \geq \gamma_\lambda \text{ AND} \\ \left|\Delta_{\mathbf{p}} - \Delta_{\mathbf{p}(\lambda_1)}\right| > \gamma_\Delta \end{array}\right)$ **then**
  $\lambda \leftarrow \text{binarySearch}\left(\lambda, \lambda_1, \Delta_{\mathbf{p}(\lambda)}, \Delta_{\mathbf{p}(\lambda_1)}, \mathcal{G}, W, \Delta, u^o, u^d\right)$
end if

References

[1] T. Gnann, T. S. Stephens, Z. Lin, P. Plötz, C. Liu, J. Brokate, What drives the market for plug-in electric vehicles ? - A review of international PEV market diffusion models, Renewable and Sustainable Energy Reviews 93 (2018) 158–164. doi:10.1016/j.rser.2018.03.055.
URL https://doi.org/10.1016/j.rser.2018.03.055

[2] Driving into 2025: The Future of Electric Vehicles, Tech. rep., J.P. Morgan (2018).
URL https://www.jpmorgan.com/global/research/electric-vehicles

[3] A. Sciarretta, A. Vahidi, Energy-Efficient Driving of Road Vehicles - Toward Cooperative, Connected, and Automated Mobility, Springer International Publishing, 2019.

[4] K. Boriboonsomsin, M. J. Barth, W. Zhu, A. Vu, Eco-Routing Navigation System Based on Multisource Historical and Real-Time Traffic Information, IEEE Transactions on Intelligent Transportation Systems 13 (4) (2012) 1694–1704.

[5] Y. M. Nie, Q. Li, An Eco-Routing Model Considering Microscopic Vehicle Operating Conditions, Transportation Research Part B 55 (2013) 154–170.

[6] J. Wang, A. Elbery, H. A. Rakha, A real-time vehicle-specific eco-routing model for on-board navigation applications capturing transient vehicle behavior, Transportation Research Part C 104 (2019) 1–21. doi:10.1016/j.trc.2019.04.017.

[7] S. Kluge, C. Santa, S. Dangl, S. Wild, M. Brokate, K. Reif, F. Busch, On the Computation of the Energy-Optimal Route Dependent on the Traffic Load in Ingolstadt, Transportation Research Part C 36 (2013) 97–115.

[8] M. Masikos, K. Demestichas, E. Adamopoulou, M. Theologou, Machine-learning methodology for energy efficient routing, IET Intelligent Transport Systems 8 (3) (2014) 255–265. doi:10.1049/iet-its.2013.0006.

[9] C. Sun, X. Hu, S. J. Moura, F. Sun, Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles, IEEE Transactions on Control Systems Technology 23 (3) (2015) 1197–1204. doi:10.1109/TCST.2014.2359176.

[10] L. Taccari, Integer programming formulations for the elementary shortest path problem, European Journal of Operational Research 252 (2016) 122–130.

[11] J. E. Beasley, N. Christofides, An Algorithm for the Resource Constrained Shortest Path Problem, Networks 19 (4) (1989) 379–394.

[12] L. Di Puglia Pugliese, F. Guerriero, A Survey of Resource Constrained Shortest Path Problems : Exact Solution Approaches, Networks 62 (3) (2013) 183–200.

[13] P. Cao, T. Miwa, T. Morikawa, Use of Probe Vehicle Data to Determine Joint Probability Distributions of Vehicle Location and Speed on an Arterial Road, Transportation Research Record 2421 (1). doi:10.3141/2421-12.

[14] R. Liu, X. Zhu, Statistical Characteristics of Driver Accelerating Behavior and Its Probability Model, arXiv:1907.01747.

[15] K. R. Bouwman, T. H. Pham, S. Wilkins, T. Hofman, Predictive Energy Management Strategy Including Traffic Flow Data for Hybrid Electric Vehicles, IFAC PapersOnLine 50 (1) (2017) 10046–10051. doi:10.1016/j.ifacol.2017.08.1775.
URL https://doi.org/10.1016/j.ifacol.2017.08.1775

[16] G. Wu, K. Boriboonsomsin, M. J. Barth, S. Member, Development and Evaluation of an Intelligent Energy-Management Strategy for Plug-in Hybrid Electric Vehicles, IEEE Transactions on Intelligent Transportation Systems 15 (3) (2014) 1091–1100.

[17] D. Karbowski, V. Sokolov, A. Rousseau, Vehicle Energy Management Optimization through Digital Maps and Connectivity, in: ITS World Congress, 2015.

[18] M. Montazeri-Gh, M. Mahmoodi-K, Optimized predictive energy management of plug-in hybrid electric vehicle based on traffic condition, Journal of Cleaner Production 139 (2016) 935–948. doi:10.1016/j.jclepro.2016.07.203.
URL http://dx.doi.org/10.1016/j.jclepro.2016.07.203

[19] A. Le Rhun, F. Bonnans, G. De Nunzio, T. Leroy, P. Martinon, A stochastic data-based traffic model applied to vehicles energy consumption estimation, IEEE Transactions on Intelligent Transportation Systems (2019) 1–10doi:10.1109/TITS.2019.2923292.

[20] S. J. Moura, H. K. Fathy, D. S. Callaway, J. L. Stein, A Stochastic Optimal Control Approach for Power Management in Plug-In Hybrid Electric Vehicles, IEEE Transactions on Control Systems Technology 19 (3) (2011) 545–555. doi:10.1109/TCST.2010.2043736.

[21] X. Jiao, T. Shen, SDP Policy Iteration-Based Energy Management Strategy Using Traffic Information for Commuter Hybrid Electric Vehicles, Energies 7 (7) (2014) 4648–4675. doi:10.3390/en7074648.

[22] X. Fuguo, J. Yuan, J. Xiaohong, A Modified Energy Management Strategy Based on SDP Policy Iteration for Commuter Hybrid Electric Vehicles, in: 2016 35th Chinese Control Conference (CCC), TCCT, 2016, pp. 2537–2541. doi:10.1109/ChiCC.2016.7553745.

[23] G. Scora, M. Barth, Comprehensive modal emission model (cmem) version 3.01 user's guide, University of California Riverside Center for Environmental Research and Technology 23 (2006) 24.

[24] H. Rakha, K. Ahn, A. Trani, Development of VT-Micro Model for Estimating Hot Stabilized Light Duty Vehicle and Truck Emissions, Transportation Research Part D 9 (1) (2004) 1–44.

[25] C. De Cauwer, W. Verbeke, J. Van Mierlo, T. Coosemans, A Model for Range Estimation and Energy-Efficient Routing of Electric Vehicles in Real-World Conditions, IEEE Transactions on Intelligent Transportation Systems.

[26] L. Guzzella, A. Sciarretta, Vehicle Propulsion Systems, Springer-Verlag Berlin Heidelberg, 2013.

[27] A. Sciarretta, G. De Nunzio, L. L. Ojeda, Optimal Ecodriving Control: Energy-Efficient Driving of Road Vehicles as an Optimal Control Problem, IEEE Control Systems Magazine 35 (5) (2015) 71–90. doi:10.1109/MCS.2015.2449688.

[28] G. De Nunzio, A. Sciarretta, I. Ben Gharbia, L. L. Ojeda, A Constrained Eco-Routing Strategy for Hybrid Electric Vehicles Based on Semi-Analytical Energy Management, in: IEEE 21st Conference on Intelligent Transportation Systems, 2018, pp. 355–361.

[29] A. Houshmand, C. G. Cassandras, Eco-Routing of Plug-In Hybrid Electric Vehicles in Transportation Networks, in: IEEE 21st Conference on Intelligent Transportation Systems, 2018, pp. 1508–1513. arXiv:arXiv:1810.01443v1.

[30] M. Salazar, A. Houshmand, C. G. Cassandras, M. Pavone, Optimal Routing and Energy Management Strategies for Plug-in Hybrid Electric Vehicles, in: IEEE 22nd Conference on Intelligent Transportation Systems, 2019.

[31] M. M. Nejad, L. Mashayekhy, D. Grosu, Optimal Routing for Plug-in Hybrid Electric Vehicles, Transportation Science 51 (4) (2017) 1304–1325.

[32] R. Hassin, Approximation schemes for the restricted shortest path problem, Mathematics of Operations Research 17 (1) (1992) 36–42.

[33] M. Strehler, S. Merting, C. Schwan, Energy-efficient shortest routes for electric and hybrid vehicles, Transportation Research Part B 103 (2017) 111–135.

[34] S. Pourazarm, C. G. Cassandras, T. Wang, Optimal routing and charging of energy-limited vehicles in traffic networks, International Journal of Robust and Nonlinear Control 26 (2016) 1325–1350.

[35] G. De Nunzio, L. Thibault, A. Sciarretta, Bi-Objective Eco-Routing in Large Urban Road Networks, in: IEEE 20h International Conference on Intelligent Transportation Systems, 2017, pp. 57–63.

[36] R. J. Allgor, P. I. Barton, Mixed-Integer Dynamic Optimization, Computers & Chemical Engineering 21 (1997) S451–S456.

[37] A. Flores-Tlacuahuac, L. T. Biegler, Simultaneous mixed-integer dynamic optimization for integrated design and control, Computers & Chemical Engineering 31 (2007) 588–600. doi:10.1016/j.compchemeng.2006.08.010.

[38] R. T. Marler, J. S. Arora, The Weighted Sum Method for Multi-Objective Optimization : New Insights, Structural and Multidisciplinary Optimization 41 (6) (2010) 853–862.

[39] R. Bellman, On a Routing Problem, Quarterly of Applied Mathematics 16 (1) (1958) 87–90.

[40] P. M. Castro, Tightening piecewise McCormick relaxations for bilinear problems, Computers and Chemical Engineering 72 (2015) 300–311.

[41] R. Widyono, The Design and Evaluation of Routing Algorithms for Real-time Channels, Tech. rep., Tenet Group (1994).

[42] X. Yuan, On the extended Bellman-Ford algorithm to solve two-constrained quality of service routing problems, in: International Conference on Computer Communications and Networks, 1999, pp. 304–310.

[43] F. Morlock, B. Rolle, M. Bauer, O. Sawodny, Time Optimal Routing of Electric Vehicles Under Consideration of Available Charging Infrastructure and a Detailed Consumption Model, IEEE Transactions on Intelligent Transportation Systems.

[44] D. O'Connor, Notes on the Bellman-Ford-Moore Shortest Path Algorithm and its Implementation in MATLAB, Tech. rep., Dublin University College (2012).

[45] G. De Nunzio, L. Thibault, Energy-Optimal Driving Range Prediction for Electric Vehicles, in: IEEE Intelligent Vehicles Symposium, 2017, pp. 1608–1613.