A Time- and Energy-Optimal Routing Strategy for Electric Vehicles with Charging Constraints

Giovanni De Nunzio, Ibtihel Ben Gharbia, Antonio Sciarretta

***Abstract*— Accurate long-distance route planners for electric vehicles could help to alleviate driving range anxiety. A time- and energy-optimal routing strategy with consideration of battery charging constraints is presented here. The accuracy of the approach is improved by modeling the impact of weather and traffic conditions on the vehicle's energy consumption, as well as by considering realistic charging functions. The routing problem is cast as a multi-objective optimization and solved with different algorithms to assess the accuracy and the tractability of each method. Results show that appealing trade-offs in terms of trip time and energy consumption appear when speed becomes a decision variable and is adjusted on some portions of the route.**

***Index Terms*— Electric vehicles, charge scheduling, driving range, constrained shortest path, eco-routing.**

I. Introduction

Studies show that a small-sized electric vehicle (EV) with a roughly 100-mile range could meet the energy requirements of about 90% of the driving days, in different cities across the US [1]. The needs of most single-car households would be met while requiring behavioral modification on no more than 5% of days [2]. Yet, when considering EVs for personal use, driving range anxiety is arguably an inevitable thought. To overcome this problem and to prevent such a fear from slowing down EVs sales and diffusion, both proprietary or manufacturers' route planners (e.g. `tesla.com/trips`) and third-party or enthusiasts' platforms (e.g. `abetterrouteplanner.com`) are nowadays spreading rapidly. The objective of these services is to both reassure the drivers about the dissemination of charging stations, and suggest smart routes and charging sequences over long-distance trips to reach the final destination with minimum time waste. However, the accuracy of such route suggestions is highly dependent on the modeling framework of the system under analysis (i.e. road network, traffic and weather conditions, expected driver's behavior, vehicle dynamics, charging functions, etc.). Furthermore, it is often limiting to optimize only one performance criterion at a time, and multi-objective optimization may suggest the existence of interesting trade-offs for the final user.

In this work, the problem of finding a time- and energy-optimal route for an EV requiring multiple battery charging events to complete a long-distance trip is addressed. The proposed routing strategy is capable of providing also the optimal sequence of charging events, as well as a speed recommendation on highway road stretches to reduce charging needs. The approach consists in two computation stages. The offline part is dedicated to the preprocessing and storage of detailed topographical information of the considered routing area of interest. The online stage is entitled of converting the user input in terms of origin, destination, departure time and initial battery state-of-charge (SoC) into a multi-objective optimization problem on a weighted routing graph. Such an optimization could be framed as a constrained shortest-path (CSP) problem, which is known to be NP-hard [3]. To cope with this intractability, heuristics and ε-approximations have been proposed to achieve a solution, which may perform well in certain scenarios but, in the most general case, they cannot provide any performance guarantee and act unpredictably [4]. Promising polynomial-time approximation schemes for CSP algorithms have been proposed [5] by performing costs rounding and scaling and listing only Pareto-optimal sub-paths. This same idea has been applied to the search of optimal paths in graphs for EVs [6], [7] with SoC constraints and charging events consideration. These approximation schemes consist in reducing the number of sub-paths that the standard CSP algorithm would store at each relaxation step by rounding the value of a cost (e.g. time) 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 [6]. The same approximation could be also applied to the value of a resource (e.g. energy) to further reduce the number of retained sub-paths [7]. Evidently, such techniques give approximate solutions, and the level of approximation, as well as computational complexity, is determined by the choice of the parameter ε, which is not intuitive to tune. Other approaches, on the other hand, attempt to reduce computational complexity by breaking down the general problem in several cascaded simpler optimizations [8], [9], or by reducing the algorithm search space by means of heuristics [10], [11]. However, all these works fail to show the accuracy and the attractiveness of the routing solution for the final user.

The contributions of this work can be summarized as follows. First, the accuracy of trip time and energy demand prediction is increased by using realistic battery charging functions, as well as traffic and weather dependent powertrain modeling. In particular, novel probabilistic features have been integrated into a driver's model in order to predict major speed disruptions along the route. Second, a comparison between two CSP algorithms has been carried out to highlight the accuracy of the routing solution. In particular, it is shown

G. De Nunzio, I. Ben Gharbia and A. Sciarretta, IFP Energies nouvelles, 1 et 4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France, {giovanni.de-nunzio, ibtihel.ben-gharbia, antonio.sciarretta}@ifpen.fr

that graph-expansion techniques offer enhanced accuracy and robustness as compared to ε-approximation methods.

The paper is structured as follows. Section II describes the static and dynamic road network data used for system modeling. The routing graph definition and expansion for consideration of charging events and traveling speed as additional decision variables are detailed in Section III. The optimization problem is defined in Section IV. Section V describes the two compared algorithms. Simulation results and computation time analysis are presented in Section VI.

II. Road network data

A. Road topography

The topography of the considered road network, including road grade, infrastructure and signalization, may be assumed to be time-invariant, or varying over time with much slower dynamics than the considered problem. Therefore, these data can be stored offline on local databases for faster online route calculation. In order to populate such an offline database, REST (representational state transfer) web services of popular mapping and location data providers are often used. In this work HERE Maps was used. Let us introduce the set of road topography data used throughout this work as $\mathcal{T}_\iota$, for all road segments ι in the considered network $\mathcal{X}$. The road network discretization is proper to each data provider, and the length ℓ_ι of each road elementary unit can vary from few meters to several hundreds of meters. Each road segment ι is typically categorized by its importance, with an associated "functional" class $\mathcal{C}_\iota \in [1,5]$, ranging from $C_\iota = 1$ for major highways, to $C_\iota = 5$, for secondary urban streets. Furthermore, each road segment is defined by a unique identifier $\mathcal{N}_\iota$ and by its geometry as an n-by-3 matrix $\mathcal{S}_\iota$, where $n \geq 2$ is the number of three-dimensional coordinates available for segment ι (at least the coordinates of the two ends of the segment are provided). Coordinates are expressed in latitude, longitude and altitude above sea level. Finally, the type of intersection and/or signalization at the downstream end of the road segment can be retrieved and denoted as a vector of booleans $\mathcal{I}_\iota \in \mathbb{R}^k$, where k is the number of selected types of intersection. In this work, $k = 5$ and $\mathcal{I}_\iota$ comprises traffic lights, stops signs, yield signs, roundabouts and toll booths. Therefore, the road topography information can be defined by the set

$$\mathcal{T} = \bigcup_{\iota \in \mathcal{X}} \mathcal{T}_\iota = \bigcup_{\iota \in \mathcal{X}} \{\ell_\iota, \mathcal{C}_\iota, \mathcal{N}_\iota, \mathcal{S}_\iota, \mathcal{I}_\iota\}. \tag{1}$$

B. Charging infrastructure

The location and the available power of the charging stations within the considered network $\mathcal{X}$ are key parameters to schedule the optimal sequence of charging events. A plethora of REST web services, either based on community contributions or on OEMs data, provide this kind of information. In this work, the REST API (Application Programming Interface) of OpenChargeMap, a non-commercial crowd-sourced service, was chosen for its completeness and clarity of information. Considering the problem of long-distance routing and charge planning for EVs, it is safe to assume that a driver would only choose rapid charging points along the route in compliance with his/her trade-off between trip time and overall energy consumption. Therefore, the set of only rapid DC (direct current) chargers φ, with information about location coordinates Λ_φ and available power P_φ, is defined as:

$$\mathcal{R} = \bigcup_{\varphi \in \mathcal{X}} \mathcal{R}_\varphi = \{\Lambda_\varphi, P_\varphi\}. \tag{2}$$

The use of any "slower" charger would likely provide a trip-time dominated (in the sense of Pareto) routing solution, thus less appealing to the final user. Note that aging of the battery, due to rapid charge, is neglected in this study. Also, all the charging points in the offline database are assumed to be available. An upper supervisory layer in charge of determining the available infrastructure and assuring the recharge could come from an external provider and is absolutely compatible with the proposed approach.

C. Dynamic traffic information

When the user specifies its desired origin and destination, along with the desired departure time, any existing REST routing API may be used to retrieve the time-optimal route options, up to a specified number κ of alternatives, according to the current traffic conditions. In this work, the HERE Maps routing API was used, but it is important to notice that the provided route alternatives are not considered by the proposed approach as such. Only the suggested road segments individually and the real-time road information on the provided routes are employed to construct a routing graph. In other words, the final route suggestion calculated by the proposed approach can be different from the ones obtained from the employed routing service. Clearly, this choice of searching the optimal route among the available road segments suggested by the routing API translates into an approximated solution. The real optimum might be on a different route, absent from the retrieved alternatives. However, for long-distance trips, this is quite unlikely due to the driver's preference for faster routes and highways, as also discussed in [7].

Furthermore, in order to correctly estimate the impact of weather conditions and ambient temperature on vehicle's energy consumption (as discussed later in the paper), a REST weather API is used to retrieve current conditions at departure and forecast them along the route at fixed time (e.g. every hour) or space (e.g. every 60 km) intervals. Interpolation methods are then used to obtain weather information on every road segment. The Dark Sky API was used here.

Therefore, in analogy with the road topography data, let us introduce the set of dynamic traffic data $\mathcal{D}_i$, for all road segments i in the set of suggested route alternatives $\mathbf{X} \subset \mathcal{X}$. The traffic information is typically provided in terms of average speed of traffic $\bar{V}_i$ and average trip time $\bar{T}_i$ to travel on road segment i. The average travel speed on the next segment $\bar{V}_{i+1}$ is also provided. The ambient temperature Θ_i is obtained for each road segment i by interpolating the temperature forecast obtained from the weather API. The unique segment identifier N_i is used to retrieve the previously

stored topographical information $\mathcal{T}_i$, since $i \in \mathbf{X} \subset \mathcal{X}$. Finally, the type of intersection and/or signalization stored in $\mathcal{I}_i \in \mathcal{T}_i$ is used in the online stage to determine the probability of stopping (i.e. vehicle's speed transition to zero) at the corresponding infrastructure element. This probability of stopping supersedes the average traffic speed $\bar{V}_i$ and is used in the driver model (as discussed later in the paper) to refine the predicted driving behavior. Such a simple probabilistic model can be obtained via a classification method relating the stopping behavior with the type of intersection, the functional class of the considered road segment, and the time of the day to capture the impact of peak and off-peak traffic.

The features vector ξ used as an input of the prediction model is defined for every training example m as:

$$\xi = \left[h_i^{(m)}, \mathcal{C}_i^{(m)}, \mathcal{I}_{i,k}^{(m)} \right], \tag{3}$$

where $h_i^{(m)}$ denotes the time of day of example m on segment i, $\mathcal{C}_i^{(m)}$ denotes the functional class of segment i for example m, $\mathcal{I}_{i,k}^{(m)}$ with $k = \{1, 2, 3, 4\}$ is a vector of features indicating the presence of a traffic light, a stop sign, a yield sign, or a roundabout, respectively. Note that the toll booths were voluntarily kept out of the features vector because of their limited presence in the training data-set. In the following, the probability of stopping at a toll booth is considered the same as the one of stopping at a stop sign.

The vector of output labels for every training example m is defined as:

$$y = v_{10,i}^{(m)}, \tag{4}$$

and it consists of a boolean variable indicating whether a speed below 10 km/h was observed on each of the recorded speed profiles in the second half of the road segment i. Logistic regression was used for classification and the model was trained on a data-set of real-world driving data (i.e. floating car data, or FCD) recorded in the Greater Paris and Lyon area (France), consisting of approximately 200 thousand road segments and over 2 million data samples, or observations (i.e. driving profiles). This model achieved a prediction accuracy of 86.6% on the used training data-set (ξ, y), and is employed online to retrieve the discrete probability density function p_i of stopping or not (i.e. binary random variable) at the end of segment i as a function of the features ξ.

Finally, the dynamic traffic data can be defined as:

$$\mathcal{D} = \bigcup_{i \in \mathbf{X}} \mathcal{D}_i = \bigcup_{i \in \mathbf{X}} \left\{ \bar{T}_i, \bar{V}_i, \bar{V}_{i+1}, \Theta_i, N_i, N_{i+1}, p_i \right\}. \tag{5}$$

III. Routing graph

The definition of the routing graph takes place in the online stage of the proposed strategy because the considered road network is solely composed by the segments of the retrieved route alternatives $i \in \mathbf{X} \subset \mathcal{X}$. Such a road network can be naturally modeled as a directed graph, and it is important to notice that this graph is acyclic, thanks to the particular structure of $\mathbf{X}$. Let $\mathcal{G} = (V, A)$ be such a graph, where V is the set of vertices (i.e. the ends of each road segment i), and A is the set of arcs (i.e. road segments i connecting the nodes of the graph).

In any route planning strategy, the underlying optimization problem has an obvious decision variable, which is the feasible (according to network connectivity) path to go from a specified origin to a destination. In this work, since a long-distance route planner for EVs with charging capabilities is considered, additional decision variables should be included. This is achieved by expanding graph $\mathcal{G}$ in a directed multigraph (or multidigraph), that is by creating additional arcs where multiple decisions are possible.

A. Travel-speed multigraph expansion

The first additional decision variable allows the route planner to choose the travel speed to adopt on certain road segments along the route in order to find the right trade-off between trip time and energy expenditure. In other words, on certain high-speed portions of the route, it might be worth traveling at a lower speed than the average traffic speed in order to reduce energy consumption, and therefore reduce the amount of energy required to complete the trip, which may ultimately result in reducing the number of stops to charge the battery. The high-speed road segments are identified by $\bar{V}_i$ being higher than a certain threshold, and thus denoted as $i_v \in A$. Speed as a decision variable is added in the multidigraph $\mathcal{G}' = (V, A')$ by creating as many copies of the arcs i_v as the desired levels of travel speed below the actual traffic speed $\bar{V}_i$. In this work, the allowed speed reduction is defined as $\delta_v \in \{0, 5, 10, 15\}$ km/h below the actual speed. Therefore, the arcs $i_v \in \mathcal{G}$ become $i_{v,\delta_v} \in \mathcal{G}'$, for all possible predefined values of δ_v.

B. EV-charging multigraph expansion

The second additional decision variable allows the optimizer to choose at which charging point $\varphi \in \mathcal{X}$ to charge along the route and how much energy to recover in the battery during charge. All the charging points within a certain radius r from the vertices of graph $\mathcal{G}$ are identified, thus denoting the charging vertices $u_c \in V$. Therefore, all the outgoing arcs from the charging vertices are denoted as $i_c \in A$. The multidigraph $\mathcal{G}'$ is then further expanded to contain as many copies of the arcs i_c as the predefined levels of battery energy recovery during charge. In this work, the allowed levels of charge are defined as $\delta_c \in \{0, 10, 20, \ldots, 100\}\%$, with a step of 10%. Therefore, the arcs $i_c \in \mathcal{G}$ become $i_{c,\delta_c} \in \mathcal{G}'$, for all possible predefined values of δ_c.

C. Line graph construction

Since the driving behavior on an arc depends on the characteristics of the following arc, namely on the transition between $\bar{V}_i$ and $\bar{V}_{i+1}$, as discussed in the following, the multidigraph $\mathcal{G}'$ can be conveniently transformed in a line graph $\mathcal{L}(\mathcal{G})$ [12]. Let us recall that the line graph $\mathcal{L}(\mathcal{G}) = (A', A^*)$ of the directed graph $\mathcal{G}' = (V, A')$ has a vertex for each arc in $\mathcal{G}'$ and each arc represents a pair of adjacent arcs in $\mathcal{G}'$.

IV. Routing problem definition

The routing graph $\mathcal{L}(\mathcal{G})$ obtained with the previous expansions allows the route optimization problem to correctly account for charging and travel speed options along the route. In order for the route planner to be able to suggest the optimal route in terms of trip time and energy consumption, the graph needs to be weighted with accurate cost estimations. In the following, let us present the models to compute time and energy required to travel on each arc. Then, the constrained optimization problem to find the optimal route is formulated for the weighted routing graph.

A. Trip time

The time required to traverse a road segment depends on the type of arc $i \in A'$. The travel time estimate on an arc i where no additional decisions in terms of speed and charging are possible is provided by the routing API according to the current traffic conditions and available in $\bar{T}_i \in \mathcal{D}_i$. On the arcs i_{v,δ_v} where the speed is reduced with respect to the traffic speed $\bar{V}_i$, the travel time increases with respect to $\bar{T}_i$ and can be calculated as follows:

$$\bar{T}_{i,v,\delta_v} = \bar{T}_i + \delta_v \frac{\ell_i}{\bar{V}_i(\bar{V}_i - \delta_v)}. \tag{6}$$

Furthermore, on the arcs i_{c,δ_c} where a charge event occurs, the travel time estimate must take into account several contributions as follows:

$$\bar{T}_{i,c,\delta_c} = \begin{cases} \bar{T}_i, & \text{if } \delta_c = 0 \\ \bar{T}_i + T_d + T_s + T_c, & \text{if } \delta_c > 0. \end{cases} \tag{7}$$

The first term $\bar{T}_i$ could be replaced by $\bar{T}_{i,v,\delta_v}$ if on the arc there is a multiple-speed option, as discussed before. The second term T_d is the detour time required to reach the off-route charger location, and it depends on the distance from the closest vertex of the routing graph and on an assumed average detour speed v_d. The third term T_s is constant and represents the time spent after stopping the vehicle to interact with the charger and set-up the charge. Note that in (7), the fourth term T_c corresponds to the actual charge time, which depends on the battery SoC at the beginning of the charge, the final SoC, the battery capacity and the charger power. Since the SoC at which the vehicle begins the charge is an output of the route optimization, this contribution to the total travel time on the arc is added by the routing algorithm during the problem solution, as discussed later. In order to correctly estimate the charging time T_c, a simple model inspired by the CC-CV (constant-current constant-voltage) charging method, with a pre-charge phase when the battery capacity is low, was used in this work [13]. As an illustrative example, Figure 1 shows the charging time required for a battery with a capacity of 30 kWh and an available charger power ranging from 50 kW to 150 kW. It appears that the charging function is nonlinear and the most time-convenient charging operation arises when the battery SoC is between roughly 30% and 70%, which corresponds to the constant-current phase.

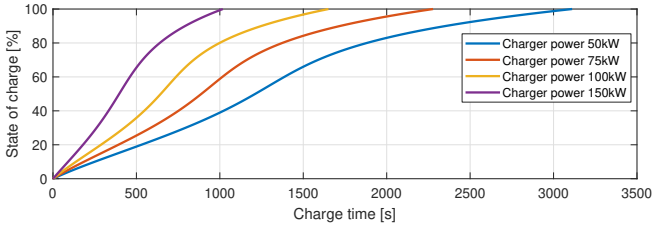


Fig. 1: Rapid charging time profile for a battery with a capacity of 30 kWh and different charger powers.

To resume, let us define a generic trip-time weight for each arc of the routing graph $\mathcal{L}(\mathcal{G})$ as:

$$\omega_{i,t} = \begin{cases} \bar{T}_{i,v,\delta_v}, & \text{if } i \in i_{v,\delta_v} \\ \bar{T}_{i,c,\delta_c}, & \text{if } i \in i_{c,\delta_c} \\ \bar{T}_{i,v,\delta_v} + T_d + T_s + T_c, & \text{if } i \in i_{v,\delta_v} \cap i_{c,\delta_c},\ \delta_c > 0 \\ \bar{T}_i, & \text{otherwise.} \end{cases} \tag{8}$$

B. Energy consumption

1) Driver's model: In order to accurately estimate the energy consumption required for traveling on an arc of the routing graph, the first step consists in predicting the driver's behavior on the road segment, so to predict the vehicle's traction power demand. In this work, the Intelligent Driver Model (IDM) [14] is used to generate expected synthetic speed profiles on each arc of the routing graph $\mathcal{L}(\mathcal{G})$, depending on the dynamic traffic data $\mathcal{D}_i$. Note that the IDM is a car-following model describing the position $x(t)$ and velocity $v(t)$ dynamics of vehicles interacting with preceding vehicles. In a route planning framework, it is not possible to precisely know the behavior of the vehicles preceding the ego-vehicle (i.e. the vehicle for which the route is optimized). Therefore, the original IDM was slightly adapted to be used for a vehicle without interaction with real preceding vehicles. When a road infrastructure element $\mathcal{I}_i$ is present on road segment i, and the realization of the random variable whose probability density is given by p_i is such that the vehicle is supposed to stop at the end of the segment, then a virtual leading vehicle appears to perturb the ego-vehicle's speed and forces a speed transition to zero. On the contrary, when no stopping behavior arises, the ego-vehicle targets unperturbed the next segment's average speed. Let us now introduce the IDM equations used in this work. The position dynamics of the ego-vehicle is simply written as:

$$\dot{x}(t) = v(t). \tag{9}$$

The speed dynamics depends on the boundary conditions on the considered road segment, and in particular on the presence of a stopping event at the upstream and/or downstream end of the segment. If no stopping event is predicted at the downstream end of the segment, only the free-road term of the IDM appears in the speed dynamics:

$$\dot{v}(t) = a\left(1 - \left(\frac{v(t)}{V_T}\right)^{\gamma}\right) = \dot{v}_{\text{free}}. \tag{10}$$

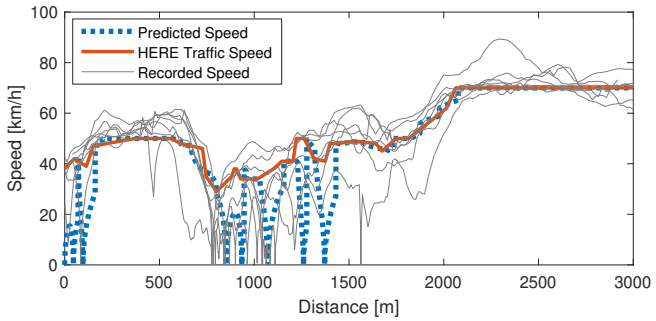


Fig. 2: Example of speed profile prediction via the modified IDM and the probabilistic stopping behavior model as compared to the HERE API traffic information and the real-world recorded speed profiles.

where a is the maximum vehicle acceleration, γ can be seen as a driver's responsiveness parameter ($\gamma = 3$ in this work), and V_T is the target speed of the ego-vehicle. In this work, the target speed is set depending on the length of the road segment and the position of the ego-vehicle as follows:

$$V_T = \begin{cases} \bar{V}_i, & \text{if } x(t) < \max\{0, \ell_i - d_h\} \\ \bar{V}_{i+1}, & \text{if } x(t) \geq \max\{0, \ell_i - d_h\}. \end{cases} \quad (11)$$

where d_h is the length of the driver's prediction horizon, a tunable parameter, which denotes the distance from which the driver starts to target the following segment's speed. Furthermore, the initial speed $v(0)$ can be set equal to 0 if the there is a stopping event at the end of the upstream segment, and equal to $\bar{V}_i$ otherwise.

On the other hand, if a stopping event is predicted at the downstream end of the segment, the speed dynamics includes also an interaction term and a virtual leader is supposed to influence the behavior of the ego-vehicle, as follows:

$$\dot{v}(t) = \dot{v}_{\text{free}} - a\left(\frac{d_0 + v(t)T_h}{x(t)} + \frac{v(t)(v(t) - v_l)}{2\sqrt{ab}\,x(t)}\right)^2 \quad (12)$$

where d_0 is the minimum inter-distance from the virtual leader, T_h is the minimum time headway from the leader, and b is a comfortable braking acceleration. The leader's speed v_l is calculated as follows:

$$v_l = \begin{cases} \bar{V}_{i+1}, & \text{if } x(t) < \max\{0, \ell_i - d_h\} \\ 0, & \text{if } x(t) \geq \max\{0, \ell_i - d_h\}. \end{cases} \quad (13)$$

If the leader is stopped (i.e. $v_l = 0$), its stop location is at the end of the segment plus the extra distance due to d_0 and the length of the vehicle itself, so that the ego-vehicle can stop exactly at the end of the segment.

An illustrative example of the speed profile prediction obtained with the modified IDM is shown in Figure 2. Note that the proposed probabilistic IDM-inspired model allows the vehicle to come to a full stop at the locations where it is likely that infrastructure and signalization would significantly perturb the speed profile. This is in contrast with other models that provide only average traffic speed, or an average synthetic profile out of a pool of recorded speed profiles.

2) Vehicle model: The energy consumption of the electric vehicle is calculated in this study by means of its longitudinal dynamics and powertrain modeling for correct approximation of the electric drive losses and auxiliary power absorption. The traction force at the wheels for a given speed $v(t)$ (i.e. the speed provided by the modified IDM) is defined as [15]:

$$F_w(t) = m\dot{v}(t) + c_2 v(t)^2 + c_1 v(t) + c_0 + mg\sin(\alpha), \quad (14)$$

where m is the vehicle mass, g is the gravitational acceleration, α is the road slope varying along the road segment as given by $\mathcal{S}_i$, and the coefficients c_0, c_1 and c_2 are identified for a considered vehicle. The wheel force (14) can be converted in the mechanical power requested from the propulsion system:

$$P_m(t) = F_w(t)v(t) \cdot \eta_t^{-\operatorname{sign}(F_w(t))}, \quad (15)$$

where η_t is the transmission efficiency. Finally, the battery energy consumption is defined as:

$$E_b = \int_0^{t_f} P_m(t) \cdot \eta_b^{-\operatorname{sign}(P_m(t))} + P_{\text{aux}}(t),\, \mathrm{d}t \quad (16)$$

where η_b represents the electric-drive efficiency, t_f is the time to traverse the road segment, and P_{aux} is the auxiliary power absorption along the segment. In this work, the auxiliary power demand is assumed to come primarily from cabin thermal conditioning, or driver's comfort needs, and is a convex function of temperature Θ_i, available on each road segment i, as in [16], [7].

To resume, let us define a generic energy consumption weight for each arc of the routing graph $\mathcal{L}(\mathcal{G})$ as:

$$\omega_{i,e} = \begin{cases} E_b - \delta_c C, & \text{if } i \in i_{c,\delta_c},\, \delta_c > 0 \\ E_b, & \text{otherwise,} \end{cases} \quad (17)$$

where C denotes the battery capacity.

C. Problem formulation

In the following, the route planning problem is formulated as a bi-objective optimization aiming to find the best trade-off between trip time and energy consumption. Let us denote with ζ the arcs of the routing graph $\mathcal{L}(\mathcal{G})$, with i^+ the set of arcs ζ entering $i \in A'$, and with i^- the set of arcs ζ leaving $i \in A'$. Let also i^o be the origin arc and i^d the destination arc. Finally, let us denote with $\mathcal{P}_i$ the sub-path composed by all the arcs ζ connecting the origin i^o to i. The route optimization problem can be formulated as follows:

$$\begin{cases} \min\limits_{x_\zeta} & \sum\limits_{\zeta \in A^*} (\lambda\omega_{i,t} + (1-\lambda)\omega_{i,e}) \cdot x_\zeta & \text{(18a)} \\ \text{s.t.} & \sum\limits_{\zeta \in i^+} x_\zeta - \sum\limits_{\zeta \in i^-} x_\zeta = \begin{cases} 1, & \text{if } i = i^o \\ -1, & \text{if } i = i^d \\ 0, & \text{otherwise} \end{cases} & \text{(18b)} \\ & C_{\min} \leq \sum\limits_{\zeta \in \mathcal{P}_i} \omega_{i,e} \leq C,\ \forall i \in A' & \text{(18c)} \\ & x_\zeta \in \{0, 1\}, & \text{(18d)} \end{cases}$$

where the objective function (18a) is written as a weighted sum of the time and energy costs on each arc of the routing

graph, with λ being the trade-off weight. The decision variable x_ζ takes on binary values as imposed by (18d) depending on whether the arc ζ belongs to the path or not. Constraints (18b) are classical flow conservation constraints. Constraint (18c) enforces every possible sub-path to verify the physical bounds of battery capacity. Note that, in order to reduce range anxiety, the minimum battery state-of-charge $C_{\min}$ can be greater than zero.

V. Routing algorithm

The objective function in (18a) can take on negative values because of the presence of the energy term, therefore two variations of the known Bellman-Ford (BF) shortest-path algorithm are proposed to solve the defined dynamic program.

A. BF algorithm on the lexicographic product of graphs

The first approach consists in resolving the constraint (18c) by enforcing it directly in the construction of the routing graph. This method, also known in the literature of routing for EVs as "battery graph expansion" [6], performs the lexicographic product of the routing graph and a fully-disconnected graph representing the feasible battery levels that can be used in the optimization. In particular, let such a graph be $\mathcal{B} = (V_b, A_b)$, where $V_b = \{0, \delta_b, 2\delta_b, \dots, C\}$ represents the battery discretization capacity from 0 to maximum capacity with a step of δ_b, and $A_b = \emptyset$. Then, the new routing graph $\mathcal{G}_b$ integrating the battery constraint is the result of the lexicographic product $\mathcal{L}(\mathcal{G}) \cdot \mathcal{B}$, where the vertex set of $\mathcal{G}_b$ is given by the cartesian product $A' \times V_b$, and any two vertices (u, u_b) and (w, w_b) are adjacent if the vertices $u, w \in A'$ are adjacent, with $u_b, w_b \in V_b$ indicating the initial and final SoC on the arc. Note that at this point, during the lexicographic product, it is possible to correctly calculate the charging time T_c on each arc knowing u_b and w_b. It is evident that the size of the new routing graph $\mathcal{G}_b$ can increase significantly if the battery discretization step δ_b is small, which is generally a requirement for solution accuracy. This reduces the scalability of the approach to larger road networks. However, while the price to pay in graph memory occupancy is high, the advantage of this approach is that a polynomial-time algorithm, such as the unconstrained BF, can be used to find the optimal solution. In the following, we simply denote with LBF the Bellman-Ford algorithm with early termination condition (i.e. the iteration on the nodes is stopped when no new relaxation on the arcs is performed) presented in [17], [12] and applied here to the lexicographic product of graphs.

B. ε-approximation constrained BF algorithm

The ε-approximation idea for the constrained Bellman-Ford (CBF) algorithm proposed here was inspired by [6], [7], and is denoted in the following as ε-ACBF. In particular, the strict Pareto-optimality condition for the acceptance of a new sub-path is dropped because, in presence of charging capabilities, a sub-path can still be globally optimal even though it is dominated in terms of time or energy by another sub-path at an intermediate charging location. In other words, when the vehicle reaches a charging location along the route, its trip time or energy consumption can be higher than another possible sub-path (e.g. charge already performed at an earlier location), and still be a candidate for the global optimal route up to destination. Therefore, let us recall that the ε-ACBF can be run on the routing graph $\mathcal{L}(\mathcal{G})$, and let us define the conditions for acceptance of candidate sub-path $\mathcal{P}_u^*$, reaching $u \in A'$, in the set of already stored sub-paths $\mathcal{P}_u$, $\forall\, u \in A'$.

The first condition aims to verify sub-path feasibility, that is the sub-path cumulated energy consumption does not exceed the physical battery boundaries:

$$\mathcal{P}_u = \mathcal{P}_u \cup \mathcal{P}_u^*, \iff \sum_{\mathcal{P}_u^*} \omega_{i,e} \in [0, C]\,. \tag{19}$$

The second condition implements the ε-approximation to reduce the time-complexity of the algorithm:

$$\begin{aligned} \mathcal{P}_u = \mathcal{P}_u \cup \mathcal{P}_u^*, \iff & \left| \sum_{\mathcal{P}_u^*} \omega_{i,e} - \sum_{\mathcal{P}_{u,\beta}} \omega_{i,e} \right| \geq \varepsilon_e \wedge \\ \wedge & \left| \sum_{\mathcal{P}_u^*} \omega_{i,t} - \sum_{\mathcal{P}_{u,\beta}} \omega_{i,t} \right| \geq \varepsilon_t, \quad \forall\, \mathcal{P}_{u,\beta} \in \mathcal{P}_u \neq \mathcal{P}_u^*, \end{aligned} \tag{20}$$

where the positive parameters ε_e and ε_t denote the minimum cost difference in terms of energy and time, respectively, for the candidate sub-path $\mathcal{P}_u^*$ to be added to the list $\mathcal{P}_u$. Note that in this case, while the graph memory occupancy is low, the algorithm is non-polynomial and the size of the stored paths can grow exponentially.

VI. Simulation Results

The first set of simulation results aims to show the higher accuracy of the LBF algorithm as compared to the ε-ACBF algorithm. To do so, we considered an example of trip from Paris to Dijon (France) with at least one charging event necessary to complete the trip (the assumed battery capacity is 30 kWh). The two algorithms were used to find the approximated optimal solution of problem (18a-18d), and compared to a reference algorithm able to find the exact solution. Such a reference is represented by a variation of the classical (CBF) algorithm [18], which was modified to store all the battery-feasible intermediate paths and not only the ones improving the cost. For $\lambda = 1$, the three algorithms calculate the same route. However, while LBF was able to calculate exactly the same sequence of charging events as CBF, ε-ACBF obtained a different solution, as summarized in Table I. The ε-ACBF, for a choice of $\varepsilon_t = 10$ s and $\varepsilon_e = 10$ Wh, gives a solution with a lower energy consumption, a lower number of charging stops and a higher trip time. However, this solution is inaccurate because there actually exists a solution with a lower trip time. Note that the average CPU time of LBF is composed by the BF run time plus the additional time required to perform the lexicographic product of graphs (i.e. 0.08 seconds in this case). The CBF algorithm was used here as a reference thanks to the reduced

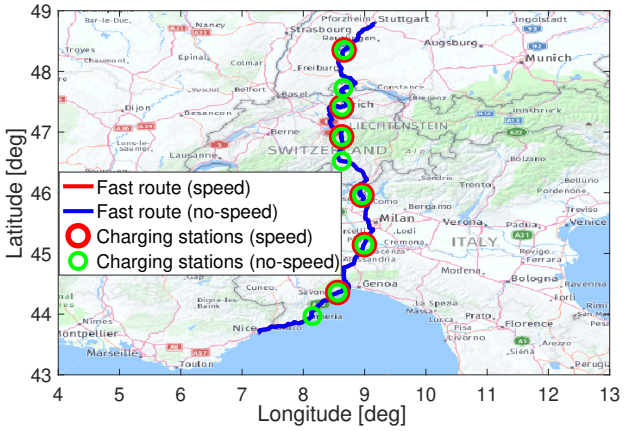
	CBF	ε-ACBF	LBF
Graph size (arcs)	140	140	397 609
Avg. CPU Time	3.22 [h]	0.03 [s]	0.01+0.08 [s]
Recharged energy [kWh]	48	48	48
Energy consumption [kWh]	74.78	74.53	74.78
Trip time [h]	3.53	3.86	3.53
Number of charging stops	3	2	3

TABLE I: Solution accuracy of LBF and ε-ACBF algorithms

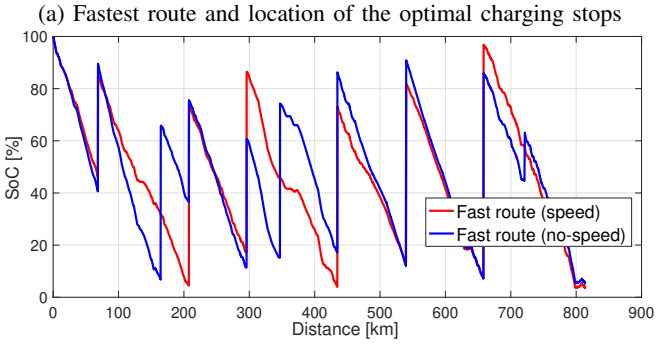
graph size, but it clearly shows very poor scalability and its time complexity grows exponentially. The algorithms were implemented in MATLAB on a computer with CPU Intel(R) Core(TM) i7-8850H CPU at 2.6 GHz and 16 GB of RAM.

The second set of simulations aims to analyze the optimal solution on a long-distance route, and in particular the impact of traveling speed as a decision variable on the solution, as well as the trade-off between energy consumption and trip time. In the following, we indicate with "speed" the case in which we use $\delta_v \in \{0, 5, 10, 15\}$ km/h, as discussed in Section III-A. We indicate with "no-speed" the case in which speed is not a decision variable, that is $\delta_v = 0$. The CBF can no longer be used as a reference because the routing graph size is already too large for this algorithm. We consider a trip from Stuttgart (Germany) to Nice (France), with traffic information on the road network provided by HERE Maps for the day of February 25, 2020, at off-peak hours.

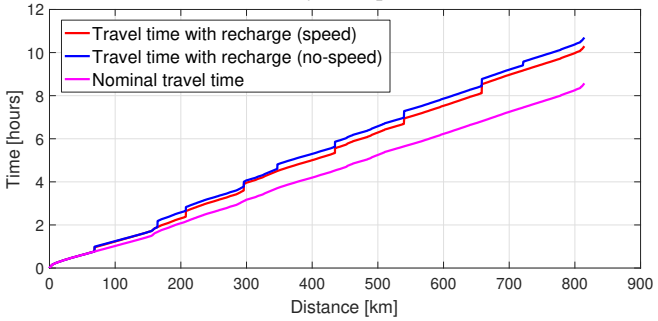
In Figure 3, the optimal solution provided by the LBF algorithm is shown in the case where the traveling speed is used as a decision variable and in the case where the vehicle is always supposed to travel at the speed of traffic. Both cases are analyzed for $\lambda = 1$, that is only the fastest route is shown. As illustrated in Figure 3a, the possibility to choose a traveling speed lower than the one of traffic does not have an impact on the fastest route itself, which remains the same in the two cases, but reduces the number of charging events necessary to complete the trip. In Figure 3b the SoC profile shows that the second charging stop was delayed thanks to a different choice of traveling speed, namely from an average speed of 120 km/h to 115 km/h between about 100 km and 200 km. This was sufficient to reduce the overall number of stops and reduce the total trip time by about half an hour, as shown in Figure 3c. The nominal travel time curve indicates the time necessary to complete the trip without any stops, which would be potentially the case with a conventional vehicle with a large enough driving range. A more detailed comparison of the results of the LBF and the ε-ACBF algorithm is given in Table II. The ε-ACBF algorithm is run now with a choice of $\varepsilon_t = 60$ s and $\varepsilon_e = 10$ Wh, given the larger size of the graph. The LBF algorithm required additional CPU time for the lexicographic product of about 0.5 s in the case without additional speeds and 17 s in the case with additional speeds. The ε-ACBF algorithm seems much faster, but it shows memory occupancy problems due to the sub-paths storage during runtime, thus reaching its scalability limits. Also, the choice of the ε parameters is such that the accuracy is very poor and the algorithm fails to show



(a) Fastest route and location of the optimal charging stops



(b) Battery SoC profiles



(c) Trip time profiles

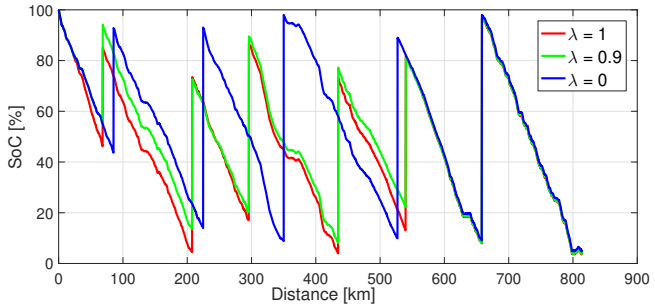
Fig. 3: Time-optimal routing solutions for a trip from Stuttgart (Germany) to Nice (France) with and without using traveling speed as a decision variable.

	No-speed		Speed	
	ε-ACBF	LBF	ε-ACBF	LBF
Graph size (arcs)	1 088	3 167 837	40 868	217 617 481
Avg. CPU Time [s]	1	0.2+0.5	37	87+17
Recharged energy [kWh]	153	153	156	123
Energy consumption [kWh]	181.5	181.5	182.7	152
Trip time [h]	11.1	10.7	12.1	10.3
Number of charging stops	9	9	14	6

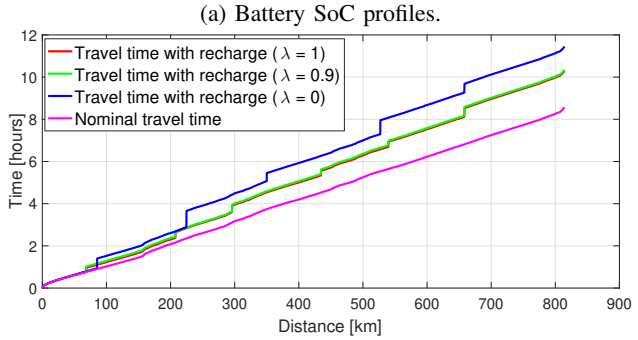
TABLE II: Performance of the routing algorithms and sensitivity to the traveling speed as a decision variable.

the appealing solution in terms of energy consumption and travel time obtained with the LBF algorithm.

Secondly, we studied the effect of the trade-off weight λ on



(a) Battery SoC profiles.



(b) Trip time profiles.

Fig. 4: Routing solution sensitivity to different values of λ.

	$\lambda = 1$ (min time)	$\lambda = 0.9$ (trade-off)	$\lambda = 0$ (min energy)
Recharged energy [kWh]	123	117	117
Energy consumption [kWh]	152	145.9	145.6
Trip time [h]	10.29	10.34	11.44
Number of charging stops	6	6	5

TABLE III: Performance sensitivity to different values of λ.

the routing solution calculated with the LBF algorithm. Table III summarizes the different results obtained for $\lambda = 0$, 0.9 and 1, while Figure 4 shows the battery SoC and trip time profiles of the different solutions. We can notice that $\lambda = 0.9$ represents a good trade-off between total energy consumption and trip time. The trip time is almost the same as the one of the fastest route (i.e. $\lambda = 1$), while the energy consumption approaches the one estimated for the eco-route (i.e. $\lambda = 0$).

VII. Conclusions

This work addresses the problem of long-distance route planning for electric vehicles with consideration of battery charging constraints, and proposes a practical implementation which aims to increase user's acceptance by combining solution accuracy and low computation time. Probabilistic features have been integrated into a driver's model in order to predict major speed disruptions along the route and improve energy consumption estimation. Two different solution algorithms are compared in order to identify the best approach to calculate the time- and energy-optimal route. The first approach (LBF) makes use of an unconstrained shortest-path algorithm on a graph expanded to integrate the battery constraints. The second approach (ε-ACBF) adopts an approximated constrained shortest-path algorithm. It is shown that the LBF algorithm offers enhanced accuracy and robustness as compared to the ε-ACBF, with comparable scalability. Finally, results show that reducing travel speed along the route may give rise to appealing solutions in terms of trip time and energy expenditure.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 824295 - CEVOLVER.

References

[1] Z. A. Needell, J. McNerney, M. T. Chang, and J. E. Trancik, "Potential for widespread electrification of personal vehicle travel in the United States," *Nature Energy*, vol. 1, no. 112, 2016.

[2] M. Khan and K. M. Kockelman, "Predicting the market potential of plug-in electric vehicles using multiday GPS data," *Energy Policy*, vol. 46, pp. 225–233, 2012.

[3] A. Artmeier, J. Haselmayr, M. Leucker, and M. Sachenbacher, "The optimal routing problem in the context of battery-powered electric vehicles," *2nd Intl. Workshop on Constraint Reasoning and Optim. for Comput. Sustainability*, 2010.

[4] F. Kuipers, A. Orda, D. Raz, and P. Van Mieghem, "A comparison of exact and e-approximation algorithms for constrained routing," *Lecture Notes in Computer Science*, pp. 197–208, 2006.

[5] R. Hassin, "Approximation schemes for the restricted shortest path problem," *Mathematics of Operations Research*, vol. 17, no. 1, pp. 36–42, 1992.

[6] M. Strehler, S. Merting, and C. Schwan, "Energy-efficient shortest routes for electric and hybrid vehicles," *Transportation Research Part B*, vol. 103, pp. 111–135, 2017.

[7] F. Morlock, B. Rolle, M. Bauer, and 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*, 2019.

[8] F. Baouche, R. Billot, R. Trigui, and N. E. El Faouzi, "Electric Vehicle Green Routing with Possible En-Route Recharging," in *IEEE 17th International Conference on Intelligent Transportation Systems*. IEEE, 2014, pp. 2787–2792.

[9] T. Wang, C. G. Cassandras, and S. Pourazarm, *Energy-aware vehicle routing in networks with charging nodes*, 2014, vol. 19, no. 3.

[10] T. M. Sweda, I. S. Dolinskaya, and D. Klabjan, "Adaptive routing and recharging policies for electric vehicles," *Transportation Science*, vol. 51, no. 4, pp. 1326–1348, 2017.

[11] C. Liu, M. Zhou, J. Wu, C. Long, and Y. Wang, "Electric Vehicles En-Route Charging Navigation Systems: Joint Charging and Routing Optimization," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 906–914, 2019.

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

[13] F. A. V. Pinto, L. H. M. K. Costa, and M. D. de Amorim, "Modeling Spare Capacity Reuse in EV Charging Stations based on the Li-ion Battery Profile," in *International Conference on Connected Vehicles and Expo (ICCVE)*, 2014, pp. 92–98.

[14] M. Treiber, A. Hennecke, and D. Helbing, "Congested Traffic States in Empirical Observations and Microscopic Simulations," *Physical Review E*, vol. 62, pp. 1805–1824, 2000.

[15] A. Sciarretta, G. De Nunzio, and L. L. Ojeda, "Optimal Ecodriving Control: Energy-Efficient Driving of Road Vehicles as an Optimal Control Problem," *IEEE Control Systems Magazine*, vol. 35, no. 5, pp. 71–90, 2015.

[16] G. De Nunzio, L. Thibault, and A. Sciarretta, "Model-based eco-routing strategy for electric vehicles in large urban networks," in *SpringerBriefs in Applied Sciences and Technology*, 2017, pp. 81–99.

[17] J. Y. Yen, "An Algorithm for Finding Shortest Routes from All Source Nodes to a given Destination in General Networks," *Quarterly of Applied Mathematics*, vol. 27, pp. 526–530, 1970.

[18] R. Widyono, "The Design and Evaluation of Routing Algorithms for Real-time Channels," Tenet Group, Tech. Rep., 1994.