IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 October 2023, accepted 17 October 2023, date of publication 20 November 2023, date of current version 29 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3334392

RESEARCH ARTICLE

Charge-Arrival-Time Profiles for Long EV Routes

SIMONAS ŠALTENIS, ANDRIUS BARAUSKAS, AGNĖ BRILINGAITĖ, LINAS BUKAUSKAS, (Member, IEEE), VAIDA ČEIKUTĖ, AND ALMINAS ČIVILIS
Institute of Computer Science, Vilnius University, 08303 Vilnius, Lithuania

Corresponding author: Simonas Šaltenis (simonas.saltenis@mif.vu.lt)

This work was supported by the European Regional Development Fund under a grant agreement with the Research Council of Lithuania (LMTLT) under Project 01.2.2-LMT-K-718-02-0018.

ABSTRACT Electric vehicles (EVs) with increasingly large batteries and the rapid development of charging infrastructure enable convenient long-distance electric mobility. Nevertheless, as charging or waiting for an available charger may still take a significant fraction of the travel time, planning long routes is challenging. We argue that, for the plans to be useful, the main challenge lies not in computing an optimal route under some specific assumptions but in managing the inherent uncertainty. In particular, varying and uncertain traffic results not only in similarly uncertain travel times but also in uncertain energy use and, thus, uncertain required charging times. To model long EV routes, we propose *Charge-Arrival-Time* (CAT) profiles. A CAT profile of a route captures the expected intervals and associated probabilities of arrival time and the arrival charge level of the battery as well as any interdependence of the two. A rich underlying data model is used as a starting point for a stepwise presentation of the mathematical model of CAT profiles. A heuristic algorithm to select charging stops is presented to exemplify their use. Detailed simulations on the road network of Germany show the benefits of precise modeling compared to alternative models.

INDEX TERMS Electric vehicle, long-distance EV routing, charge-arrival-time profiles, energy consumption, first-in-first-out, time dependency, data modeling.

I. INTRODUCTION

Influenced by many factors, the electrification of transport is accelerating. For example, the EU plans to ban all non-zero-emission vehicles starting from 2035 [1]. The renewal of the global fleet of vehicles is accompanied by the gradual development of the necessary charging infrastructure. Nevertheless, the availability of this infrastructure varies both spatially and temporally. Thus, planning long EV trips with charging stops currently involves a lot of uncertainty. The uncertainty stems from many factors. Time-varying traffic affects travel times as well as energy use, which, for EVs, is highly sensitive to speed. Energy use, in turn, determines charging-station reachability and charging duration. Charging duration can also involve waiting at busy charging stations.

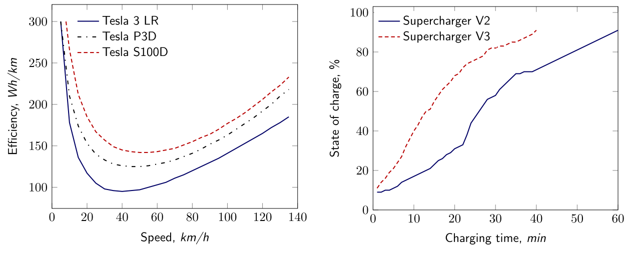
Managing this uncertainty is often important. For example, quantifying the probability of arriving before a given deadline may be relevant in the case of delivering goods or following a given schedule.

We argue that with the increased availability of collected data and learned prediction models, careful data modeling is of paramount importance.

The most advanced existing work on algorithms for EV routing [2] does a careful modeling of charging (see Figure 1b), but driving is often modeled somewhat simplistically. Either it is assumed that network edges are assigned a given constant travel time and energy use or a range of travel times and corresponding energy usages are attached to each edge. On the other hand, real-world navigation services, such as Google Maps, use time-dependent road networks, where the time it takes to traverse a given network edge or a path depends on the time the traversal starts. This time dependence is crucial, especially for longer trips and especially for electric vehicles, where energy consumption is greatly affected by travel speed.

There is a large body of work on routing in time-dependent road networks [3]. Most of the work assumes a simple model,

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco Perez-Pinal.



(a) energy use efficiency

(b) charging curve

FIGURE 1. Tesla charging and energy consumption functions (adapted from [6], [7]).

where the time of arrival at the starting vertex of an edge determines an expected time of traversal of the edge. Such an expected time can be computed from models based on historical data. The real traversal time will, of course, most probably differ from the expected time. The further into the future a system tries to predict, the more inaccurate it will likely be. That is why recently stochastic, time-dependent edge weights were proposed [4], [5]. In the simplest of such models, the likely time interval of traversal for a given edge or a path is modeled. To accommodate EVs, energy-use intervals could be similarly modeled as well.

In this paper, we argue that decoupled modeling of travel time and energy use does not capture the strong inverse interdependence of these two variables. For example, computing a conservative total trip time, one could take the slowest travel times on the edges of the trip and add the charging time derived from the highest energy use on the trip. Though simple, this may be too conservative, as the slowest travel times do not imply the highest energy use. Instead, as Figure 1 shows, slow travel times may imply efficient energy use. Specifically, going 80 km/h in a tight traffic, when compared to cruising at 140 km/h, may save about 15 kWh per 300 km of travel. This, in turn, translates to increased range with a given battery or to 15 minutes of saved charging time on a 60 kW charger.

Given a route, possibly with charging stops, we aim for a model that would enable queries such as: *What is the probability that the vehicle will arrive at the destination not later than 5 p.m.? If it arrives by that time, what would the expected range of charge remaining in the battery be?*

To address these data-modeling challenges, we propose *Charge-Arrival-Time (CAT) profiles*. To the best of our knowledge, this is the first data model that, for routes with multiple charging stops, 1) assumes uncertain, time-dependent travel times and energy use, 2) captures interdependence between travel time and energy use, and 3) captures charging curves and time-dependent uncertain waiting times at chargers. A heuristic algorithm to select charging stops is presented to exemplify the use of CAT profiles. The paper also contributes with an extended simulation study on a German road network. The study evaluates the use of CAT profiles in contrast to alternative, more limited models.

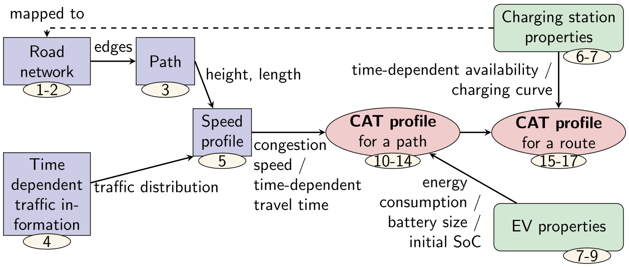

mapped to
Road network
1-2
edges
Path
3
height, length
Speed profile
5
Time dependent traffic information
4
traffic distribution
congestion speed / time-dependent travel time
CAT profile for a path
10-14
CAT profile for a route
15-17
Charging station properties
6-7
time-dependent availability / charging curve
energy consumption / battery size / initial SoC
EV properties
7-9

FIGURE 2. A roadmap of the proposed model.

The paper is structured as follows. Section II, in a sequence of steps, formally defines and explains CAT profiles. As a case study for the possible use of CAT profiles, Section III presents a simple heuristic algorithm for choosing and evaluating charging stops on a given route. The experimental setup and results are described in Section IV. Section V reviews related work. The paper ends up with conclusions and future work in Section VI.

II. MODELING LONG EV ROUTES

As presented in the introduction, our goal is to model arrival times and arrival charge levels on long EV trips with charging stops. In the following, we build up a formal model, precisely defining a set of concepts. We do it step by step, combining the following building blocks as shown in Figure 2: the (relatively) static road-network infrastructure; dynamic, time-dependent traffic information; EV-specific information; and charging-station-specific information. The arrows in the figure point from simpler concepts or groups of concepts to more complex concepts that build on them. The arrows are annotated with sub-concepts that are relevant when defining the more complex concepts. Finally, the numbers refer to the definitions later in this section.

Table 1 summarizes the main notations used in the following.

A. ROAD-NETWORK GRAPH

Definition 1: A *road network* is modeled as a directed graph $G = (V, E)$. An edge $e \in E$ is a 4-tuple $e = (u, v, l, hp)$, where $u, v \in V$ are the start and the end vertices of the edge, $l \in \mathbb{R}_{\geq 0}$ represents the length of the edge and $hp \in \mathcal{HP}$ is a *height profile* of an edge which includes all edge-specific information necessary to compute energy consumption on the edge.

Note that hp implicitly includes the length of the edge, but we represent it separately to simplify the presentation.

Definition 2: Charging station type function $cs : V \rightarrow \mathcal{CST} \cup \{\perp\}$ returns the type of a charging station, $cst \in \mathcal{CST}$, available at a given vertex $v \in V$. The set $\mathcal{CST}$ represents charging station types, each of which defines a type of a plug and an available power level. If no charging station is available at v, then $cs(v) = \perp$.

TABLE 1. Summary of the main notations.

Notation	Meaning
$e = (u, v)$	A graph edge from vertex u to vertex v
P; P^i	A path in a graph; its i-th edge
R; P_i; R^{-1}	A route; its i-th leg; R without the last leg
$\overline{t} = [t^{\vdash}, t^{\dashv}]$; $\overline{c} = [c^{\vdash}, c^{\dashv}]$	Intervals of time and charge, their endpoints
$tt(P, t)$; $at(P, t)$	An interval of travel/arrival time of P, leaving at t
$\mathbb{P}(\overline{t})$	Probability of arriving during $\overline{t}$
$sp(P, t)$	A speed profile of P, leaving at t
$ec(P, sp, c)$	An interval of battery charge at the end of P if starting with charge c
$ch^{-1}(cst, c_s, c_e)$	Time it takes to charge from c_s to c_e on charger type cst
$cat(EV, P, t, c)$	A CAT profile for EV driving on P, leaving at t with starting charge c
$\bigoplus S$	Convolution of a set of CAT profiles with associated probabilities (Definition 13)
$fcat(\overline{t}, c)$	A *flat* CAT profile corresponding to a departure time interval $\overline{t}$ and starting charge c (Definition 14)

Multiple types of charging stations or even one charging station with multiple types of sockets at a single charging location can be modeled by introducing several vertices connected by the *null-edges*, which are edges of zero length.

Definition 3: A *path* $P \in \mathcal{P}$ models a simple path in a road network. It is a sequence of edges $P = \langle e_1, e_2, \ldots, e_p \rangle$, where $e_i \in E$, and $e_i.v = e_{i+1}.u$ for $1 \leq i < p$. The path is required to contain no cycles.

The number of edges in path P is denoted by $|P|$ and the i-th edge of P is denoted by P^i. A height profile of P, denoted by $P.hp$, is a concatenation of the height profiles of its edges. The length of path P is denoted by $P.l = \sum_{i=1}^{|P|} P^i.l$.

B. TIME-DEPENDENT TRAFFIC INFORMATION

We assume that the system maintains a comprehensive historical traffic database. Such a database, possibly in conjunction with traffic-prediction models and current traffic information, enables the computation of time-dependent traffic predictions relevant to route planning.

First, we assume that the *time domain* $\mathcal{T}$ ($\mathcal{T} \cong \mathbb{R}_{\geq 0}$) is subdivided into a sequence of consecutive, half-open time slots of equal length: $\mathcal{T}' = \langle \tau \mid \tau = [\tau^{\vdash}, \tau^{\dashv}) \wedge \tau^{\dashv} - \tau^{\vdash} = \delta \rangle$. Thus, each time point $t \in \mathcal{T}$ belongs to a unique time slot: $t \in \tau(t) \in \mathcal{T}'$. The length of a time slot, δ, is a global system parameter determined primarily by the granularity of the collected traffic information and the employed traffic prediction models. It can range from a few minutes to as long as an hour.

Definition 4: A *travel-time function tt* : $\mathcal{P} \times \mathcal{T} \rightarrow \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$ is a function that returns the expected travel-time interval of a path $P \in \mathcal{P}$ when starting the traversal at a given time $t \in \mathcal{T}$.

In the following, we use $\overline{tt} = [tt(P, t)^{\vdash}, tt(P, t)^{\dashv}]$ to denote the returned time interval and its two ends. A similar notation is used for intervals of speed and battery charge level. Given a specific departure time t_s, the arrival time interval is computed as $at(P, t_s) = [t_s + tt(P, t_s)^{\vdash}, t_s + tt(P, t_s)^{\dashv}]$.

To simplify the representation of tt, we assume its upper and lower bounds are piece-wise linear functions, such that, in each time slot, an upper (lower) bound is represented by a single linear function.

To realistically model time-dependent travel times, we require the tt function to satisfy the FIFO property—a natural requirement that, although a vehicle A leaving at a later time than another vehicle B can take less time to traverse a path, A cannot arrive sooner than B [8], [9], [10], [11], [12]. If this property were not satisfied, waiting at some points on the path could be necessary to minimize arrival time, which contradicts common sense and reduces pruning opportunities in routing algorithms.

The FIFO property can be also extended to uncertain travel times [13], [14]:

$$\begin{aligned} \forall P:\ t_1 < t_2 < t \implies \mathbb{P}(at(P, t_1) < t) \\ \geq \mathbb{P}(at(P, t_2) < t). \end{aligned}$$

In other words, leaving later can not increase the probability of arriving before any given time t. We interpret $\overline{tt}$ as a single-bin histogram: the travel time is expected to be in the interval with a given global confidence level (say 95%) and it is assumed to be uniformly distributed inside the interval. Then, the probability of arriving before t is proportional to overlap of $[0, t]$ and $\overline{at}$ and the FIFO condition can be rewritten:

$$\begin{aligned} &\forall P:\ t_1 < t_2 \\ &\implies at(P, t_1)^{\dashv} \leq at(P, t_2)^{\dashv} \wedge at(P, t_1)^{\vdash} \leq at(P, t_2)^{\vdash} \\ &\implies t_1 + tt(P, t_1)^{\dashv} \leq t_2 + tt(P, t_2)^{\dashv} \\ &\wedge\ t_1 + tt(P, t_1)^{\vdash} \leq t_2 + tt(P, t_2)^{\vdash} \\ &\implies tt(P, t_1)^{\dashv} - tt(P, t_2)^{\dashv} \leq t_2 - t_1 \\ &\wedge\ tt(P, t_1)^{\vdash} - tt(P, t_2)^{\vdash} \leq t_2 - t_1 \\ &\implies \frac{tt(P, t_2)^{\dashv} - tt(P, t_1)^{\dashv}}{t_2 - t_1} \geq -1 \\ &\wedge\ \frac{tt(P, t_2)^{\vdash} - tt(P, t_1)^{\vdash}}{t_2 - t_1} \geq -1. \end{aligned}$$

The last two lines effectively require that the slope of the tt upper (lower) bound is not less than -1 in any interval $[t_1, t_2]$. For this to be true, we formulate two requirements:

Requirement 1: The upper (lower) bound of the travel-time interval is a continuous function.

Requirement 2: In each time slot, the upper (lower) bound of the travel-time interval is a linear function with a slope larger or equal to -1.

Example 1: Figure 3 presents an example of the travel time function, tt. Solid lines visualize its upper and lower bounds. Intervals $\overline{tt}_1$ and $\overline{tt}_2$ identify travel time when leaving at t_1 and t_2, respectively. At the top of the figure, the corresponding arrival-time intervals are presented along the x-axis. Thus, if the trip starts at $t_1 = \frac{1}{2}\delta$ and travel time is $\overline{tt}_1 = [2\delta, 3\delta]$, the arrival time is $\overline{at}_1 = [2\frac{1}{2}\delta, 3\frac{1}{2}\delta]$. The figure also shows an alternative upper and lower function

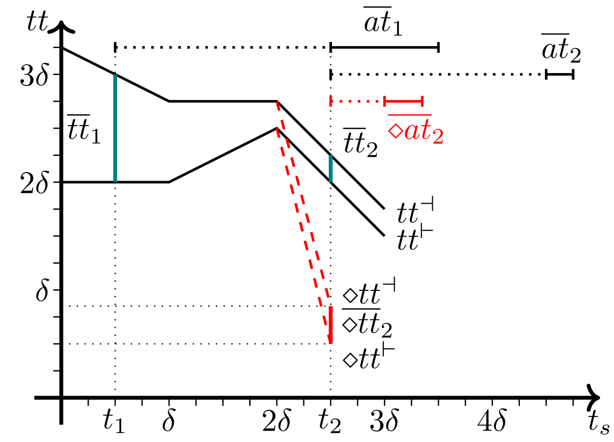


FIGURE 3. Example travel-time function with alternative, non-FIFO bounds.

bounds within time slot $[2\delta, 3\delta)$ identified by two dashed lines. For these bounds, the slope of the function is less than -1, thus, for the departure time t_2, the travel time interval is $\overline{\diamond tt}_2 = [0.5\delta, 0.85\delta]$, making the arrival time interval $\overline{\diamond at}_2 = [3\delta, 3.35\delta]$, i.e., earlier than the end of $\overline{at}_1$. Therefore, the alternative upper and lower bounds do not satisfy the FIFO property.

The travel-time interval captures the concise summary of expected traffic on a given path. It does not capture how the speed varies on different parts of the path, which can be captured by the *speed profile*.

Definition 5: A *speed-profile function sp* : $\mathcal{P} \times \mathcal{T} \rightarrow \mathcal{SP}$ captures the expected speed variation along the path when leaving at a given time.

The structure and size of the speed profile depend on the capabilities of the underlying traffic-modeling system. In the simplest case, it is just an expected interval of the average speed along the path corresponding to the expected travel-time interval: $sp(P, t) = [P.l/tt(P, t)^{\dashv}, P.l/tt(P, t)^{\vdash}]$. A richer speed profile may include individual speed intervals for subpaths of a path, e.g., each P^i.

Definition 6: Given a time point, a *waiting-time function* $wt : V \times \mathcal{T} \rightarrow \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$ returns the expected minimum and maximum time one has to wait to start charging at a charger of vertex $v \in V$. The waiting time is $[0, 0]$ if there is at least one available charger at the station.

Waiting to charge is equivalent to traveling on a zero-length edge. Thus, Requirements 1 and 2 apply to the waiting-time function as well.

Note that a charging station with a given charger type may offer different charging powers depending on how many other cars are currently being charged at the station. This can be modeled as multiple virtual charging stations with different power levels and different waiting-time functions so that the highest power level is least available (longer expected waiting time) and the lowest power level is most available (shorter expected waiting time).

C. ELECTRIC VEHICLE

A given EV is characterized by how it charges its batteries and how it consumes energy while driving.

Definition 7: A *charging function ch* : $\mathcal{CST} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is an EV-type-determined function that, given a charging station type *cst*, a starting battery charge level c_s and a time period Δt, returns the resulting battery charge level $c_e = ch(cst, c_s, \Delta t)$.

An inverse charging function $ch^{-1} : \mathcal{CST} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ computes how much time it takes to charge from one charge level to another: $\Delta t = ch^{-1}(cst, c_s, c_e)$. For example, in Figure 1b it takes 20 minutes to charge a particular battery from 8% to app. 30% and app. 70% at superchargers V2 and V3, respectively.

Definition 8: An *energy-consumption function ec* : $\mathcal{P} \times \mathcal{SP} \times \mathbb{R}_{>0} \rightarrow \mathbb{R} \times \mathbb{R}$ is an EV-type-determined function, that given a path P, a speed profile *sp* and an initial charge level c_s of the battery, returns an expected interval of charge levels of the battery after traversing the path: $\bar{c} = [c^{\vdash}, c^{\dashv}] = ec(P, sp, c_s)$.

Energy-consumption function uses the height profile of P, $P.hp$. It should also model energy recuperation on downhill parts of the height profile. Thus, it is possible that $c^{\vdash} > c_s$ and/or $c^{\dashv} > c_s$. If $c^{\vdash} < 0$, there may not be enough energy in the battery to traverse the path and we say that the given triple of the path, speed profile, and the initial charge is not *feasible*.

Real-world energy-consumption functions are markedly affected by weather conditions [15] and other context variables. To simplify, we do not model this, but this could be modeled as additional parameters of the energy consumption function.

Note that to compute *ec*, it may not be enough to check the energy consumption just at the smallest and then at the largest speeds captured in the speed profile. This is because the energy consumption function, if expressed as kWh/km, is a concave function (see Figure 1).

Definition 9: An *electric vehicle* is defined as a triple $EV = (b, ch, ec)$, where $b \in \mathbb{R}_{>0}$ is the vehicle's battery capacity, while *ch* and *ec* are its charging and energy consumption functions, respectively. Note that the values returned by *ch* and *ec* are required to be smaller than b.

D. CHARGE-ARRIVAL-TIME PROFILE

The definitions presented up to this point model the result of an EV traversing a path as an interval of the expected final charge level of the battery, coupled with an interval of the expected arrival time. The result can be thought of as a bounding-box approximation of a curve that captures how the final battery level depends on the arrival time.

Example 2: Figure 4 presents an example of a travel time function that satisfies Requirements 1 and 2. Dashed diagonal lines represent arrival times. For example, if the driver leaves at $t_s = \delta$, the lower and upper bounds of the travel function,

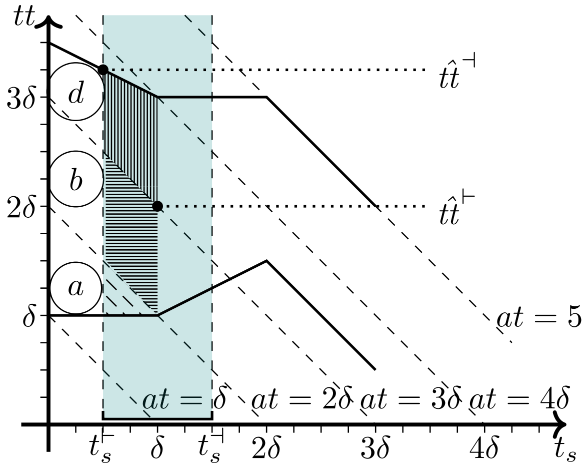


FIGURE 4. Example travel-time function.

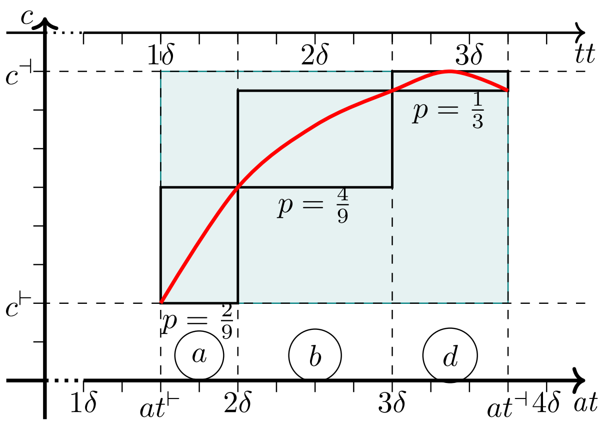


FIGURE 5. Charge-arrival-time profile for $t_s = \frac{1}{2}\delta$.

$tt^{\vdash}$ and $tt^{\dashv}$, are δ and 3δ, respectively. Therefore, the earliest and latest arrival times are 2δ and 4δ.

Figure 5 visualizes the results from Figure 4 and some *ec* function, assuming $t_s = \frac{1}{2}\delta$. The shaded rectangle illustrates the bounding-box approximation of the function shown as the curve in the figure.

Such an approximation is often too crude, particularly as the predicted rectangle becomes larger toward the end of a long route. This, in turn, will hinder decision-making for algorithms. For example, when determining the needed charging time after traversing a path, an algorithm should be able to benefit from the fact that arriving later can be partially compensated by a reduced charging time. This occurs because later arrival time translates into lower average speed, which is typically more energy-efficient, and thus may result in a fuller battery at the end of the path. In Figure 5, this is illustrated by the curve being close to $c^{\dashv}$ when the arrival time is close to $at^{\dashv}$.

To capture this interdependence more accurately, we assume that it is possible to partition the speed profile into speed intervals corresponding to subintervals of the travel-time interval.

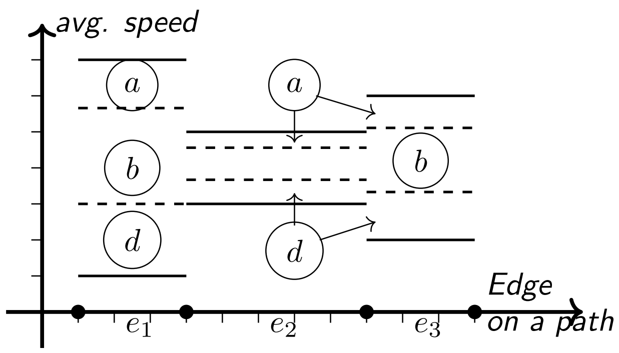


FIGURE 6. Speed profile and its partitioning.

Definition 10: Given a speed profile $sp(P, t_s)$ and a subinterval $\bar{t} \subseteq tt(P, t_s)$ of the corresponding travel-time interval, $sp_{\bar{t}}(P, t_s) \subseteq sp(P, t_s)$ denotes the *narrowing* of the speed profile corresponding to $\bar{t}$.

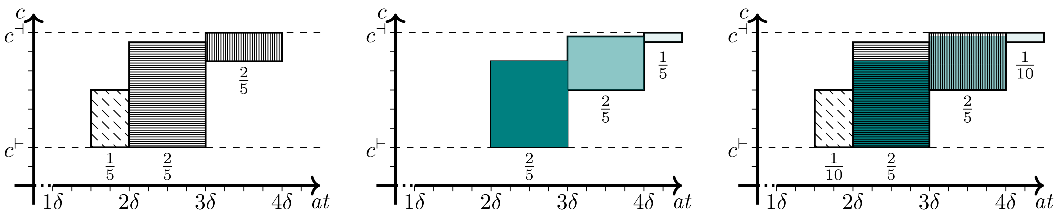
Figure 5 shows a partition of the arrival-time interval into three sub-intervals based on Figure 4. Sub-intervals are time slots or their parts that intersect the arrival-time interval. This partition then generates a corresponding partition of the travel-time interval (see the top of the figure). Finally, Figure 6 illustrates a simple way to build a corresponding partitioning of a speed profile. The intervals of average speed on each edge of the path (solid lines) are subdivided into sub-intervals of speed (dashed lines) that are proportional to the travel-time sub-intervals. For example, traveling at average speeds in intervals marked with b results in arrival sometime during the arrival-time slot b.

Definition 11: Given an electric vehicle EV, a path P, a departure time t_s, and a starting battery charge level c_s, the *charge-arrival-time (CAT) profile* of P is defined as a set of charge and arrival-time interval pairs with associated probabilities:

$$\begin{aligned} &cat(EV,\ P,\ t_s,\ c_s) \\ &\quad = \Big\{ \big(\bar{t},\ EV.ec(P,\ sp_{\bar{t}-t_s}(P, t_s),\ c_s),\ \mathbb{P}(\bar{t})\big) \mid \\ &\qquad \bar{t} = \tau \cap at(P, t_s) \wedge \bar{t} \neq \emptyset \wedge \tau \in \mathcal{T}' \Big\} \end{aligned}$$

where $\bar{t} - t_s = [t^{\vdash} - t_s, t^{\dashv} - t_s]$ is the travel-time interval corresponding to the arrival-time interval $\bar{t}$ and $\mathbb{P}(\bar{t}) = |\bar{t}|/|at(P, t_s)|$ is the probability of arriving during $\bar{t}$. Let $\mathcal{CAT}$ denote the set of all valid CAT profiles: $\mathcal{CAT} \subset 2^{\mathcal{T}\times\mathcal{T}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}_{[0,1]}}$.

A CAT profile captures the expected battery charge interval in each time slot of the arrival-time interval. Figure 5 illustrates a CAT profile as three rectangles—*elements* of the profile. Each element of the profile is associated with its probability, assuming a uniform distribution of the arrival time in the predicted arrival time interval. If some of the elements of a CAT profile have negative charge level



(a) $\overline{t_s} = [\frac{1}{2}\delta, \delta]$, $\mathbb{P} = \frac{1}{2}$ (b) $\overline{t_s} = [\delta, 1\frac{1}{2}\delta]$, $\mathbb{P} = \frac{1}{2}$ (c) $\overline{t_s} = [\frac{1}{2}\delta, 1\frac{1}{2}\delta]$

FIGURE 7. Convolution of charge-arrival-time profiles.

values, the CAT profile is called *infeasible*. For $s \in \mathcal{CAT}$, *infeasible*(s) if $\exists(\cdot, [c^{\vdash}, c^{\dashv}], \cdot) \in s : c^{\vdash} < 0$.

When multiple paths are concatenated into a route, the arrival time interval of one path becomes the departure time interval of another path. To this end, the CAT profile of a path has to be defined for an interval of departure times.

First, we look at a case when the departure-time interval is fully contained in a time slot. In Figure 4, the departure-time interval $[\frac{1}{2}\delta, 1\frac{1}{2}\delta]$ (light blue) overlaps with two time slots. Let us consider its first half $\overline{t_s} = [\frac{1}{2}\delta, \delta]$ that is fully contained in the first time slot. To build its CAT profile, for each of the time slots of the arrival-time interval, the corresponding maximal travel-time interval has to be computed. In the figure, the inclined region d contains all trips with arrival times between 3δ and 4δ. The shaded part of that region corresponds to the relevant departure-time interval. It is easy to see that the region corresponds to travel times spanning from 2δ to $3\frac{1}{4}\delta$ (see the two black dots in the figure). This interval can then be used to narrow down the speed profile and compute the corresponding element of the CAT profile. This element and the other two can be seen in Figure 7. Figure 7a shows the CAT profile for $\overline{t_s} = [\delta, 1\frac{1}{2}\delta]$.

Definition 12: Restricted travel-time function $\hat{tt}^{\vdash}(P, \overline{t_s}, \overline{t_a})$ computes the interval of possible travel times given an interval of departure times, $\overline{t_s} = [t_s^{\vdash}, t_s^{\dashv}]$, and an interval of arrival times, $\overline{t_a} = [t_a^{\vdash}, t_a^{\dashv}]$. The details of the computation of $\hat{tt}$ can be found in Appendix.

Then, formally, we augment Definition 11 to work on any $\overline{t_s} \subset \tau, \tau \in \mathcal{T}'$:

$$cat(EV, P, \overline{t_s}, c_s) \\ = \Big\{ \big(\overline{t}, EV.ec(P, sp_{\hat{tt}(P,\overline{t_s},\overline{t})}(P, \overline{t_s}), c_s), \mathbb{P}(\overline{t})\big) \mid \\ \overline{t} = \tau \cap at(P, \overline{t_s}) \wedge \overline{t} \neq \emptyset \wedge \tau \in \mathcal{T}' \Big\}.$$

Here, $\mathbb{P}(\overline{t}) = |\overline{t}|/|at(P, \overline{t_s})|$ and $at(P, \overline{t_s}) = [at(P, t_s^{\vdash})^{\vdash}, at(P, t_s^{\dashv})^{\dashv}]$. Due to the FIFO property, these two points are enough to define the arrival-time interval. The speed profile for an interval of departure times $\overline{t_s}$ is defined as a union of the speed profiles of all time points of the interval.

If a departure time interval spans multiple time slots, each time slot will have a probability associated with it. For example, if the departure time interval corresponds to the arrival time interval of the previous leg of a route, the probabilities are derived from the elements of that leg's CAT profile. To construct a CAT profile for a departure time interval, we compute a *convolution* of the probability distribution of the departure time and the time-dependent probability distribution of travel time, represented by the CAT profiles of each time slot within the departure time interval. This is achieved by "summing up" these CAT profiles. More specifically, we iterate through all the arrival time slots, and for each time slot, we compute the minimum bounding box of the relevant profile elements, as well as their probability. Figure 7a illustrates a CAT profile formed by summing the two profiles in Figures 7b and 7c. The two profiles are first constructed for the two equal halves of the departure time interval $[\frac{1}{2}\delta, 1\frac{1}{2}\delta]$, each associated with an equal probability of $\frac{1}{2}$. Then, for example, the element from 3δ to 4δ in the sum of the two profiles is a minimum bounding box of the corresponding two elements—one from Figure 7a and one from Figure 7b. The probability of this element is equal to $\frac{1}{2} \cdot \frac{2}{5} + \frac{1}{2} \cdot \frac{2}{5} = \frac{2}{5}$.

We term this operation a *convolution* of a set of CAT profiles with associated probabilities:

Definition 13 (Convolution): Given a set $S \subset \mathcal{CAT} \times \mathbb{R}_{[0,1]}$ of CAT profiles, their sum (convolution) is defined as follows

$$\bigoplus S = \Big\{ \big(\bigsqcup_{e \in S_\tau} e.\overline{t}, \bigsqcup_{e \in S_\tau} e.\overline{c}, \textstyle\sum_{e \in S_\tau} e.\mathbb{P}_e \cdot e.\mathbb{P}_{cat}\big) \mid \\ S_\tau = \big\{\langle(\overline{t}, \overline{c}, \mathbb{P}_e), \mathbb{P}_{cat}\rangle \in \bigcup S \mid \overline{t} \subseteq \tau\big\} \wedge S_\tau \neq \emptyset \wedge \tau \in \mathcal{T}' \Big\},$$

where $\bigsqcup I = [\min_{\overline{i} \in I} i^{\vdash}, \max_{\overline{i} \in I} i^{\dashv}]$ denotes the minimum bounding interval of either time or charge interval sets, and, for simplicity, it is assumed that the probability of a CAT profile in S, $\mathbb{P}_{cat}$, is associated with each element of the CAT profile. Note that the convolution of a set of CAT profiles is a CAT profile as well.

The second line of the definition takes a subset of the union of all the elements of the CAT profiles—the set S_τ that contains only the elements with arrival-time intervals in a given time slot τ. Then, the minimum bounding box of S_τ is constructed as one element of the summed CAT profile. This process is repeated for all non-empty S_τ sets.

We can now define the CAT profile for an arbitrary departure time interval. First, assuming that the departure time is uniformly distributed in the departure time interval, we construct the corresponding *flat CAT profile* that also captures the departure charge and probabilities. The profile is flat, as the lower and upper charge bounds of all the elements are equal to the departure charge. Defining the input as a CAT profile is useful when modeling multi-leg routes where a uniform distribution of the departure time can no longer be assumed starting from the second leg of a route, as described below.

Definition 14: Given a departure time interval $\overline{t_s}$ and a starting battery charge level c_s, the corresponding *flat CAT profile* is defined as follows:

$$fcat(\overline{t_s},\ c_s) = \Big\{ \big(\bar{t},\ [c_s, c_s],\ |\bar{t}|/|\overline{t_s}|\big) \mid \bar{t} = \tau \cap \overline{t_s} \\ \wedge \bar{t} \neq \emptyset \wedge \tau \in \mathcal{T}' \Big\}.$$

Then, given such a flat CAT profile *fcat*, an electric vehicle EV, and a path P, the *CAT profile* of P is defined as follows:

$$cat(EV,\ P,\ fcat) = \bigoplus \big\{ \langle cat(EV, P, \bar{t}, c_s),\ \mathbb{P} \rangle \mid \\ \big(\bar{t},\ [c_s, c_s],\ \mathbb{P}\big) \in fcat \big\}.$$

The definition computes the convolution of all the relevant CAT profiles—the cat profiles of all the departure time slots.

E. ROUTES WITH CHARGING STOPS

Given the data modeling described so far, the source vertex $v_s \in V$ and the destination vertex $v_d \in V$, our aim is to model feasible and fast routes from v_s to v_d. A route should indicate which charging stops to use, how to drive between them, and the amount of charge needed at each stop.

It is important to note that for a path to be feasible, charging at each charging stop has to be conservative, taking into account the most energy-inefficient driving on the remainder of the route, as given by the CAT profiles. On the other hand, such a conservative charging time at a given stop u will vary depending on the battery's energy level, which, in turn, depends on how the initial part of the route before u was driven. Therefore, instead of specifying the required charging time, it is more sensible to indicate the required charge level that the battery must reach at u before continuing on the rest of the route.

Definition 15: A *route* is a sequence $R = \langle (c_1, P_1), (c_2, P_2), \ldots, (c_r, P_r) \rangle$, where $P_i \in \mathcal{P}$ is a path between charging stops and $c_i \in \mathbb{R}_{>0}$ is a charge level that the battery has to be charged to at the beginning of path P_i, for $1 \leq i \leq r$. All paths, except possibly the first, start with a charging station: $cs(P_i^1.u) \neq \perp$, for $1 < i \leq r$. Finally, all consecutive paths of a route are connected to each other: $P_i^{|P_i|}.v = P_{i+1}^1.u$, for $1 \leq i < r$.

A pair of a charge level and a path constitutes a *leg* of a route and we denote R^i the i-th leg of a route. A *sub-route* from the k-th to the l-th leg of a route is denoted as $R^{k..l}$. Finally, the prefix of a route without the last leg is denoted as $R^{-1} = R^{1..|R|-1}$.

Before defining the CAT profile of a route, we first define the waiting and charging time. Both definitions involve adding a scalar to an interval, which results in the addition of the scalar to both ends of the interval: $\bar{i} + h = [i^{\vdash} + h, i^{\dashv} + h]$.

Definition 16: The function *wct* computes the waiting and charging time interval for a vehicle EV at vertex v_c to reach the charge level c_e starting from the charge level c_s if arriving within the time interval $\bar{t} \subset \tau$, $\tau \in \mathcal{T}'$:

$$wct(EV,\ v_c,\ \bar{t},\ c_s,\ c_e) \\ = \begin{cases} [0, 0] & \text{if } c_s \geq c_e, \\ \Big[wt(v_c, t^{\vdash})^{\vdash}, wt(v_c, t^{\dashv})^{\dashv}\Big] + & \\ \quad EV.ch^{-1}(cs(v_c), c_s, c_e) & \text{otherwise.} \end{cases}$$

If charging is not necessary, the function returns $[0, 0]$. Otherwise, it calculates the sum of the maximal waiting interval if arriving within $\bar{t}$ and the charging time.

Definition 17: Given an electric vehicle EV, a route $R = \langle (c_1, P_1), (c_2, P_2), \ldots, (c_{|R|}, P_{|R|}) \rangle$, a departure time t_s, and a starting battery charge level c_s, the CAT profile of R is defined as follows:

$$cat(EV,\ R,\ t_s,\ c_s) \\ = \begin{cases} cat\Big(EV, P_1, fcat\big(t_s + wct\big(EV, P_1^1, [t_s, t_s], c_s, c_1\big), c_1\big)\Big) \\ \qquad\qquad\qquad\qquad \text{if } |R| = 1, \\ cat\ \Big(EV, P_{|R|}, \bigoplus \Big\{ \big\langle fcat\big([t^{\vdash} + w^{\vdash},\ t^{\dashv} + w^{\dashv}], c_{|R|}\big),\ \mathbb{P} \big\rangle \mid \\ \quad \overline{w} = wct\ \Big(EV, P_{|R|}^1, \bar{t}, c^{\vdash}, c_{|R|}\Big) \wedge \\ \quad (\bar{t}, \bar{c}, \mathbb{P}) \in cat\big(EV, R^{-1}, t_s, c_s\big) \Big\}\Big) \qquad \text{otherwise.} \end{cases}$$

The CAT profile of a one-leg route, $|R| = 1$, is simply a CAT profile of the leg's path. However, to account for charging at the beginning, a waiting/charging-time interval is added to the departure time.

For a multi-leg route, the CAT profile is defined recursively. First, the CAT profile of the prefix of the route, R^{-1}, is computed (as seen in the last line of the definition). Then, the time interval of each element of the profile is shifted in time by adding the necessary waiting/charging-time interval $\overline{w}$ to it. Next, each such interval is converted into a flat CAT profile, dividing it into time-slot-sized elements. Finally, all these flat CAT profiles are convoluted to get the departure flat CAT profile as an input for computing the CAT profile of the last leg, which is the CAT profile of the entire route.

Figure 8 demonstrates an example of computing the departure flat CAT profile for the next leg of a route. Let the CAT profile shown in Figure 5 represent the CAT profile of a prefix of a route. The top of Figure 8 displays an example waiting-time function for the relevant time slots. The three rectangles in Figure 8 illustrate the three elements of the CAT profile shifted in time according to the waiting-time function. The values of the lower and upper bounds of the waiting-time function used for shifting are highlighted as black points on the graphs of the bounds. For example, the time interval of element b ($[2\delta, 3\delta]$) was altered by adding $\frac{1}{4}\delta$ to its beginning and $\frac{3}{4}\delta$ to its end.

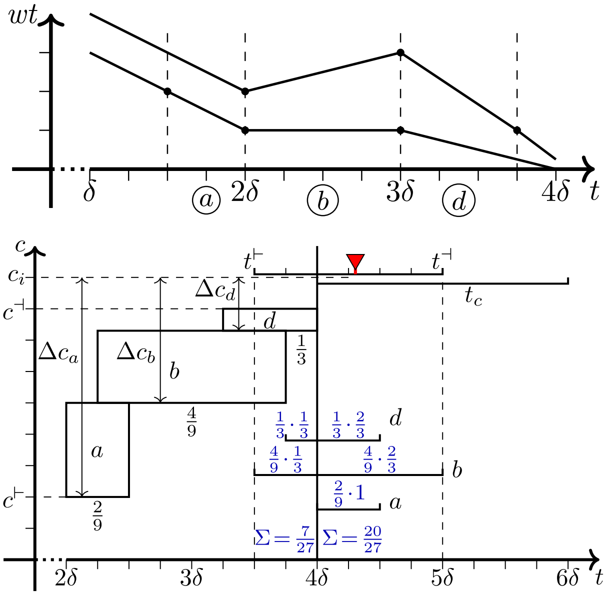


FIGURE 8. Waiting-time function and CAT profile with waiting and charging.

Figure 8 also displays the *conservative* charge intervals, equal to the difference between the charge level c_i required for the next leg and the lower end of the charge interval of the CAT profile's element. Figure 9 demonstrates how these charge intervals are then translated into the required charging times using the charging function of the vertex at the beginning of the next leg. For example, the charging time relevant to element b of the CAT profile is $\Delta t_b = \frac{5}{4}\delta$. Thus, both ends of b's time interval are shifted by that amount, resulting in an interval shown in the bottom center of Figure 8. The same is done for elements a and d of the profile. The time intervals are then converted into flat CAT profiles. As part of the conversion, they are divided into time-slot elements with probabilities proportional to the lengths of the elements. For example, the shifted b is divided into $[3\frac{1}{2}\delta, 4\delta]$ with a probability of $\frac{1}{3}$ and $[4\delta, 5\delta]$ with a probability of $\frac{2}{3}$. The shifted flat CAT profiles a, b, and d are then convoluted, and the result is shown at the top of the figure: $\overline{t} = [3\frac{1}{2}\delta, 5\delta]$. The figure also demonstrates how the probabilities of the elements of the resulting flat CAT profile are computed. For example, the element $[3\frac{1}{2}\delta, 4\delta]$ receives the probability of $\frac{1}{3}\cdot\frac{1}{3}+\frac{4}{9}\cdot\frac{1}{3}=\frac{7}{27}$.

The computed flat CAT profile t encodes the interval of departure times after charging to c_i. In addition, assuming the uniform distribution of departure times within each element of the CAT profile, the expected departure time can be computed using the probabilities of its elements. As shown by the red triangle in Figure 8, it is the weighted sum of the centers of the elements: $\frac{7}{27}\cdot 3\frac{3}{4}\delta + \frac{20}{27}\cdot 4\frac{1}{2}\delta$.

The example highlights the advantages of modeling a CAT profile as a set of profile elements rather than a single pair of arrival-time and charge intervals (as illustrated by the shaded bounding rectangle in Figure 5). As introduced in Section II-D, the more precise modeling allows for leveraging the dependencies between the arrival time and the final charge. In this example, element a of the profile represents the earliest arrival times but also requires the longest charging time before the next leg, while the opposite is true for element d. This results in the departure-time intervals of elements coalescing into a shorter departure flat CAT profile. In this example, the resulting departure-time interval $\overline{t}$ has the length of $1\frac{1}{2}\delta$. As illustrated in Figure 8, if a simple "bounding rectangle" CAT profile were used, the departure-time interval would be $\overline{t_c} = [4\delta, 6\delta]$, with the length of 2δ. In summary, fine-grained profiles with temporal granularity matching the granularity of the collected traffic data limit the growth of uncertainty in multi-leg routes.

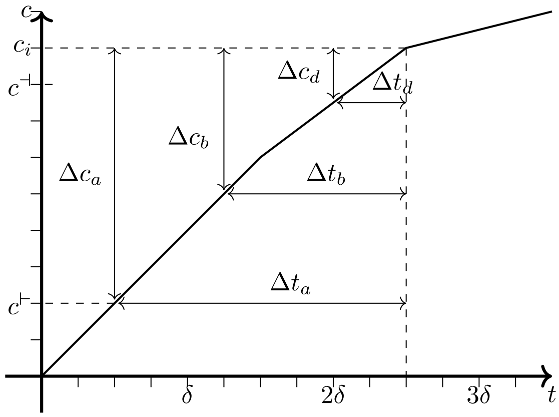


FIGURE 9. Using charging function to compute charge times.

III. USING CAT PROFILES

The development of advanced algorithms for route planning that take full advantage of the richness of CAT profiles is out of the scope of this paper. Nevertheless, in this section, we strive to demonstrate the use of CAT profiles in a context of a relatively simple heuristic algorithm.

Given a known path, computed, e.g., by existing online navigation services, the presented algorithm chooses charging stops along the path with the goal of achieving the earliest probable arrival time. The probable arrival time of a route can be computed from the CAT profile of the route given a required probability of arrival (for example, 75%). Recall that the route R is defined as a sequence of pairs (c, P) where c is the level to be charged before traversing path P (see Definition 15). We start with the given *guide* path P from source s to destination d. Parts of the calculated route, $R = \langle(c_1, P_1), \ldots, (c_{|R|}, P_{|R|})\rangle$, compose P with minor deviations due to detours to reach and depart charging stations, i.e., $P \simeq \bigsqcup_{(c_i,P_i)\in R} P_i$.

Due to the FIFO property, optimizing the probable earliest arrival time amounts to choosing charging stations and

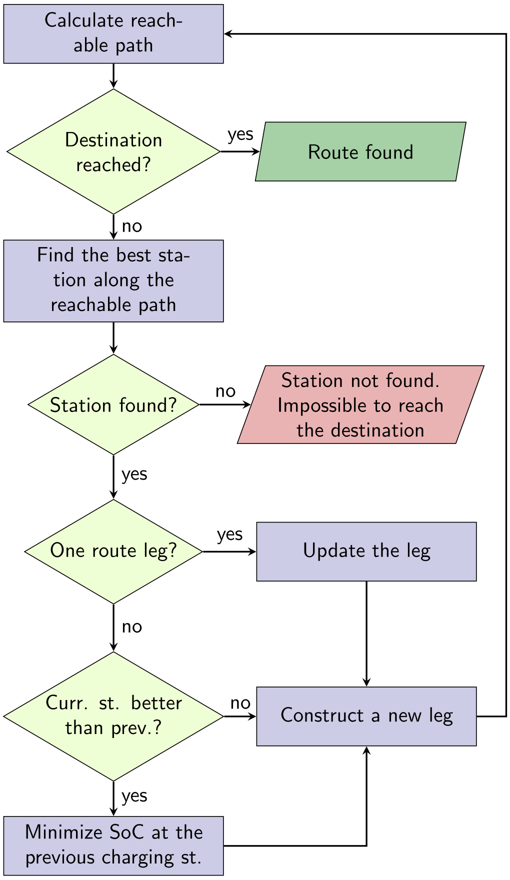

Calculate reachable path
Destination reached?
yes
Route found
no
Find the best station along the reachable path
Station found?
no
Station not found. Impossible to reach the destination
yes
One route leg?
yes
Update the leg
no
Curr. st. better than prev.?
no
Construct a new leg
yes
Minimize SoC at the previous charging st.

FIGURE 10. Flowchart diagram of the algorithm.

charge-to levels at the stations so that the total charging and waiting time at the stations is minimized. Note though that striving exclusively for the earliest arrival time may sometimes result in undersireble routes with *too many* charging stops. As described later, we introduce an algorithm parameter α to control the number of charging stops, with possible sacrifices to the arrival time.

Figure 10 presents the main steps of the algorithm as a high-level flowchart diagram. First, the reachable path is calculated. If the destination is along the reachable path, the route is found. Otherwise, charging stations along the path are identified, and the best one is selected according to a scoring function (described in the following). There could be situations when there are no charging stations, thus, the algorithm has to stop, as the destination is not reachable. If the station is found, the route can be constructed further. If the route contains only one leg, it is updated. If there are more legs, the newly found station is compared to the previous one. If the current station has a better score, the previous leg is updated to set the battery SoC to the minimum required level that would be enough to reach the newly found charging station. In both cases, a new leg is constructed using the rest of the path. The algorithm then loops to the first step to calculate the reachable path.

Algorithm 1 provides more details. The algorithm takes four inputs: EV, a guide path P, a departure time t_s, and an initial state of charge c. The algorithm then returns a calculated route, R. It employs several temporary variables: v^c and v^p denote current and previous graph nodes within the road network where charging is planned, and $P_\triangleleft$ represents a path reachable with the current state of charge.

Firstly, the algorithm initiates a route with a leg represented by the initial state of charge, c and the full path, P (step 3). Then, the algorithm searches for charging stations along the path. All charging stations are mapped to the nearest graph nodes of the road network, and charging station type function returns $\perp$ if there is no possibility of charging at a particular node (see Definition 2). In the beginning, path P represents the full path, and during iterations, it is segmented into pieces when creating route legs as the algorithm progresses. The algorithm continues until the destination becomes reachable, which is when the remaining path P falls within the reachable area (steps 5–26).

Algorithm 1 Calculation of a Heuristic Route

```
Input: $EV, P, t_s, c \in \mathbb{R}_{\geq 0}$
Output: $R$ — a suggested route
 1: Let $v^c, v^p \in V$ be nodes with a charging station
 2: Let $P_\triangleleft \in \mathcal{P}$ be a path
 3: $R \leftarrow \langle(c, P)\rangle$ {The initial SoC and pre-computed path}
 4: $R' \leftarrow \langle(c, \varnothing)\rangle$ {$R'$ is $R$ with an empty last path}
 5: repeat
 6:   $P_\triangleleft \leftarrow \text{reachPrefix}(P, EV, cat(EV, R', t_s, c))$
 7:   if $P \neq P_\triangleleft$ then
 8:     $v^p \leftarrow v^c$
 9:     $v^c \leftarrow \text{bestStation}(EV, P_\triangleleft, cat(EV, R', t_s, c))$
10:     if $v^c = \perp$ then
11:       return $\perp$
12:     end if
13:     if $|R| = 1$ then
14:       $R^1 = (c_1, P_1) \leftarrow (\cdot, P^{s,v^c})$
15:     else
16:       $R^{|R|} = (c_{|R|}, P_{|R|}) \leftarrow (\cdot, P^{v^p,v^c})$
17:       if $ecr(v^c, EV, cat^{R^{|R|}}) > ecr(v^p, EV, cat^{R^{|R|-1}})$ then
18:         $R^{|R|} = (c_{|R|}, P_{|R|}) \leftarrow (\text{minReq}(R^{-1}, R^{|R|}), \cdot)$
19:         $\text{updateCAT}(cat^{R^{|R|-1}})$
20:       end if
21:     end if
22:     $P \leftarrow P^{v^c,d}$
23:     $R \leftarrow R \sqcup (\text{mSoC}(EV, v^c), P)$
24:     $R' \leftarrow R^{|R|-1} \sqcup (\text{mSoC}(EV, v^c), \varnothing)$
25:   end if
26: until $P = P_\triangleleft$
27: return $R$
```

During each iteration, the algorithm calculates $P_\triangleleft$, the maximum prefix of P that can be reached if leaving with

EV.*b* (or *c*, if this is the first iteration) in the middle of time interval encoded in the cat of R', which is a route computed so far, up to and including the last chosen charging station. Such a prefix route can be computed by the modified energy consumption function (see Definition 8). If the remaining part of the trip does not fall into the reachable area, the algorithm searches for a charging station (steps 6–7). The previous station, v^p, is set to point to the last station at v^c, and the next suitable station, v^c, is found (9). If no such station exists, the route can't be constructed, and the algorithm returns an undefined route (11). This represents an extreme case where no charging stations can be reached with a current state of charge.

The bestStation function is used by the algorithm in step (9) to find a suitable charging station. We sketch the main idea behind this function, which finds a station that is neither too close to the start of the path nor too inefficient in charging. The function considers and scores all the charging stations located within a certain narrow spatial buffer along path $P_{\triangleleft}$. For the station at v, a heuristics-based score is calculated based on the CAT profile of the path leading to the station:

$$\begin{aligned}&score(EV, P^{\cdot,v}, cat)\\&\quad = \alpha \cdot ecr(v, EV, cat) + (1-\alpha) \cdot P^{\cdot,v}.l, \quad \text{where}\\&ecr(v, EV, cat)\\&\quad = \frac{EV.b{\cdot}mSoC - cat.c^{\vdash}}{awct(EV, v, cat.\bar{t}, cat.c^{\vdash}, EV.b{\cdot}mSoC)}.\end{aligned}$$

In the formula above, $P^{\cdot,v}$ represents the path from the previous charging station to the current one, v, while *cat* is its CAT profile. Parameter $\alpha \in [0; 1]$ is an algorithm parameter that affects a number of charging stops along the route. Smaller α values nudge the algorithm to plan fewer charging stops even at the cost of some increase in the probable arrival time. Setting α to one completely switches off the consideration of the number of charging stops.

Function *ecr* represents the *effective charging rate*, calculated by dividing the worst-case energy amount required to reach the maximum charge level *mSoC* by the value of *awct*, which is the *average waiting and charging time* at v if arriving during $cat.\bar{t}$ and charging from $cat.c^{\vdash}$ to $EV.b{\cdot}mSoC$. Due to the characteristics of their charging curves, *mSoC* of 80% is usually suggested for fast chargers.

The discovery of a new node with a charging station impacts the route legs leading to it (steps 13–21). If the route has only one leg, this leg contains the initial state of charge and the original pre-computed path. Only the path of the tuple is updated by setting it to the path P^{s,v^c} from the source s to the newly identified vertex v^c. If the route has multiple legs, the last leg of the path is updated by setting it to the path from the previous to the current station (16). Also, if the route has multiple legs, the charging rate, *ecr*, is compared at the current and previous charging stations, v^c and v^p, respectively. If *ecr* is better at the current station, then the last leg's state to charge is updated by setting it to the minimum state (18) necessary to traverse the leg $R^{|R|}$ after charging at the end of R^{-1} (route without the last leg). Moreover, the CAT profile of the last leg is also updated accordingly.

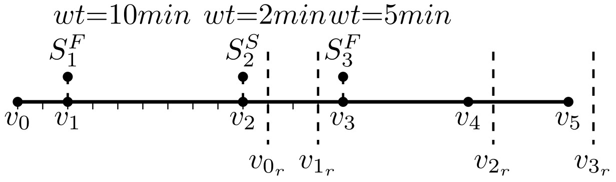


FIGURE 11. Example of route search.

After the last leg of the route is updated, the remaining path P is set to the path from the current charging station to the destination, $P^{v^c,d}$, and a new leg is added to the route (steps 22, 23). The preliminary state to charge for the new leg is set to the maximum possible level that can be achieved for *EV* at v^c, but can be updated by the algorithm later (line 18).

Example 3: Let us assume that path $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5$ is the input guide path (see Figure 11). Charging stations are denoted by S_i, and superscript identifiers F and S stand for faster and slower chargers, respectively. In the example, the power of slow charging is γ, and the power of fast charging is 3γ. Each station is mapped to the vertex of the road network and has an associated average waiting time for the relevant arrival-time window. For example, S_1^F is a faster charger with an average waiting time of 10*min*, and it is mapped to v_1. Vertical dashed lines indicate the border of the reachable area from a particular vertex with a battery level of $mSoC = 0.8$. The final reachable location from the trip source v_0 is between vertices v_2 and v_3 (see v_{0_r}), meaning *EV* will have to stop and charge at either v_1 or v_2. According to the algorithm, the stations are scored, and the optimal best is selected. Let us assume that $SoC = 0.64$ when arriving at S_1^F, and $SoC = 0.08$ when arriving at S_2^S. In this example, S_1^F has a charger that is three times faster than the one at S_2^S, but it also has five times longer average waiting time. It takes approximately *20min* and *60min* to charge an empty battery to the desired $mSoC = 0.8$ at F and S type charging station, respectively. Thus, charging at S_1^F, including waiting time, will take $4min + 10min = 14min$ and at S_2^S—$55min + 2min = 57min$. However, charging at S_1 does not provide sufficient energy to skip S_2, as the reachable area v_{1_r} is before v_3. Another option is to stop at both charging stations. After charging at S_1^F, the state of charge upon arrival at S_2^S will be $SoC = 0.24$. With this option, the total time spent at charging stations will be $14min + 45min = 59min$, which is longer than just charging at S_2^S. Thus, S_2^S is selected as the first charging stop.

Since reaching the destination v_5, after fully charging at S_2^S is impossible, the algorithm builds the route further by evaluating the next available charging stations. Station S_3^F with a three times faster charging option is further down the route. At this point, the algorithm evaluates if the battery should be fully charged at S_2^S as the faster charger S_3 is available later. The algorithm calculates how much to charge at S_2^S to have enough energy to reach S_3^F, since the charging curve isn't linear. It should be charged at S_2^S until the charging speed is faster than the minimal speed at S_3^F. Therefore it is enough to charge at S_2^S up to $SoC = 0.32$, which would only take (with waiting time) in total $18min + 2min = 20\ min$. After fully charging at S_3^F, it is possible to reach the destination, v_5, as the remaining path P^{v_3,v_5} falls within the reachable area.

Note that the presented algorithm has a linear complexity in terms of the length of the guide path. Depending on the implementation details of the underlying traffic modeling system, the main costs of the algorithm most probably lie in the computation of the CAT profiles of the legs of the computed route.

IV. EXPERIMENTS

To compare the proposed CAT profile model to simpler but less representative models, an experimental simulation study was performed. First, we briefly describe an experimental setup, followed by the presentation of the results of the experiments.

A. EXPERIMENTAL SETUP

The experiments were run on the CAT profiles implemented as a C++ simulation on top of the multi-component simulation testbed [8] that integrates various open data such as road network data, free-flow travel time data, charging station data, and daily traffic distribution data. The testbed architecture also includes a semi-synthetic data generator to prepare the missing data, such as congestion travel times and energy consumption along the road edges. Figure 12 presents a generalized software architecture. The legs of the routes used in the experiments are computed by the KaTCH router [16], which is a time-dependent earliest-arrival router implemented in C++ and based on time-dependent contraction hierarchies [17].

The data generated by the testbed generator is managed by the PostgreSQL database management system with the PostGIS extension. First, the data is used to extract a graph with time-dependent travel-time edge weights. This graph is used by the router. Next, using the generated data, the testbed provides implementations of the main functions used to compute CAT profiles: the travel-time function, tt (Definition 4), the energy-consumption function, ec (Definition 8), including its version that works on a sub-interval of the travel-time interval (Definition 10), the waiting-time function, wt (Definition 6), and the inverse charging function, ch^{-1} (Definition 7).

With these building blocks in place, the main definitions from Sections II-D and II-E were implemented as C++ classes. The source code, data, and other experimental artifacts have been made available.[1] A web-based demonstration and visualization system was also developed [18].

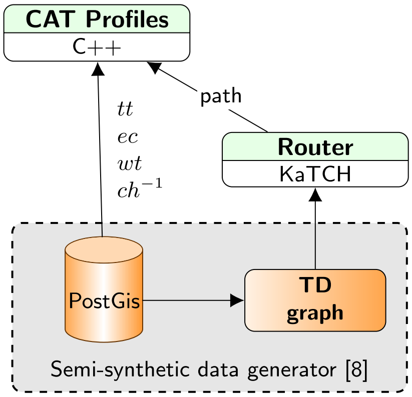


FIGURE 12. Generalized test-bed architecture (adapted from [8]).

In the following, we briefly describe the testbed and semi-synthetic data generation. Barauskas et. al [8] describe the data sources and the workings of the testbed in greater detail. The testbed uses a road network exported from OpenStreetMap [19] and keeps the essential edge data, such as type, length, direction, lane number, and speed limit. CGIAR-CSI SRTM 90m Digital Elevation Data [20], [21] was utilized to calculate the slopes on road segments and energy consumption and recuperation along the trip. Finally, data of charging stations [22] include more than 12 thousand charging stations in Germany and a number of chargers of each supported connection type for each station.

The road-network data is augmented with traffic information using the SUMO road-traffic simulator [23]. The random trip simulation follows the so-called gravity model in cities, which models that interaction between any two locations declines with increasing distance (travel time) between them, meaning that most of the trips are short and located in city centers. The number of trips decreases with increasing distance from a city center. The applied Gawron algorithm [24], [25], [26] allows achieving close to real-world behavior of driver routing. The output of the SUMO simulation is the travel time on each edge during the peak congestion hour. The testbed uses 24-hour traffic volume distribution in Germany from TomTom [27] for the so-called travel-time profile that captures time-dependent variability of travel time in between the free-flow and congestion times. The free-flow and congestion travel times are calibrated using external travel time data — Google Maps [28].

The testbed evaluates energy consumption along the route employing the Vehicle Energy Model [29], [30]. The model considers vehicle mass, time-variant vehicle speed and altitude, energy losses caused by air, rolling friction,

[1] MIDAS Open Access Research Data Archive: http://doi.org/10.18279/MIDAS.DALTRA_CAT-PROFILE.208030

curve resistance, and constant vehicle energy consumers like heaters or AC.

The testbed also provides time-dependent waiting-time intervals. They were generated based on the best charging time of the charging station, the number of connections for fast/slow charging, and the highest average speed among the edges that intersect with a predefined buffer around the charging station. As modeled in [31], in business premises, the peak of the number of arriving cars is at 8:00. The highest power demand is during 11:00–13:00. Most cars leave at 17:00. Thus, we created a power demand plan for regular working day in business premises with an increase/decrease percent for each hour. The piece-wise linear EV charging functions were implemented to approximate real sample charging curves.

For the experiments, trips between random pairs of sources and destinations were generated using the KaTCH router. Each set consists of 1000 paths, and each set groups paths of similar length: from 200 km, up to 700km. For each graph point in the experiments, 200 routes are taken randomly from all the generated routes. Next, to have a controlled experiment, the same model charger is placed artificially on each route when the battery is depleted. This is repeated up to three times along the route unless the route ends earlier.

B. EXPERIMENTAL RESULTS

To the best of our knowledge, there is no prior study that takes its outset in the similar set of modeling assumptions as this paper, thus, the goal of our experimental study is to compare the proposed CAT profiles to a simpler but intuitive baseline model under the same assumptions. We consider *bounding-boxes* that capture the expected arrival time interval and the expected arrival charge interval as such a baseline model. Such bounding boxes are not split into Δ-sized elements and there are no probabilities associated (uniformity is assumed).

1) THE METRICS

CAT profiles are essentially representations of probability distributions. Their structure is meaningful only if the said distributions are non-uniform. To measure whether that is true, the *relative entropy* (or Kullback–Leibler divergence) from the uniform distribution is calculated

$$D_{KL}(P\|Q) = \sum_{x} P(x) log\left(\frac{P(x)}{Q(x)}\right)$$

where x is a CAT profile element, $P(x)$ is its corresponding probability and $Q(x)$ is its probability if the distribution is uniform. The latter is calculated by dividing the time-interval length of the element by the time-interval length of the whole CAT profile.

The results of the experiments are presented in a grid of graphs in Figure 13. The columns correspond to the five metrics measured in the experiments. They are, from left to right:

1) *The relative entropy*, described above.
2) *The relative difference in time-interval lengths* of bounding boxes and CAT profiles. Naturally, the time intervals get longer as the uncertainty grows for longer routes with more charging stops. It is interesting to see whether there is a difference in growth between the two types of models.
3) *The relative shift of the 75% percentile of arrival time*, when compared to the same percentile under the uniformity assumption of the corresponding bounding box (the shift is relative to the length of the arrival time interval of the bounding box).
4) *The absolute shift of the said 75% percentile* in seconds. Using this metric and the previous one, we explore how the two modeling approaches answer queries about the probable arrival time. Naturally, the CAT profile is a more detailed model, but does its complexity pay off in terms of the accuracy of arrival time queries?
5) *The number of CAT elements* in the profile. Using this metric we study how fast the uncertainty grows for longer routes with more charging stops.

TABLE 2. Experimental parameters (default in bold).

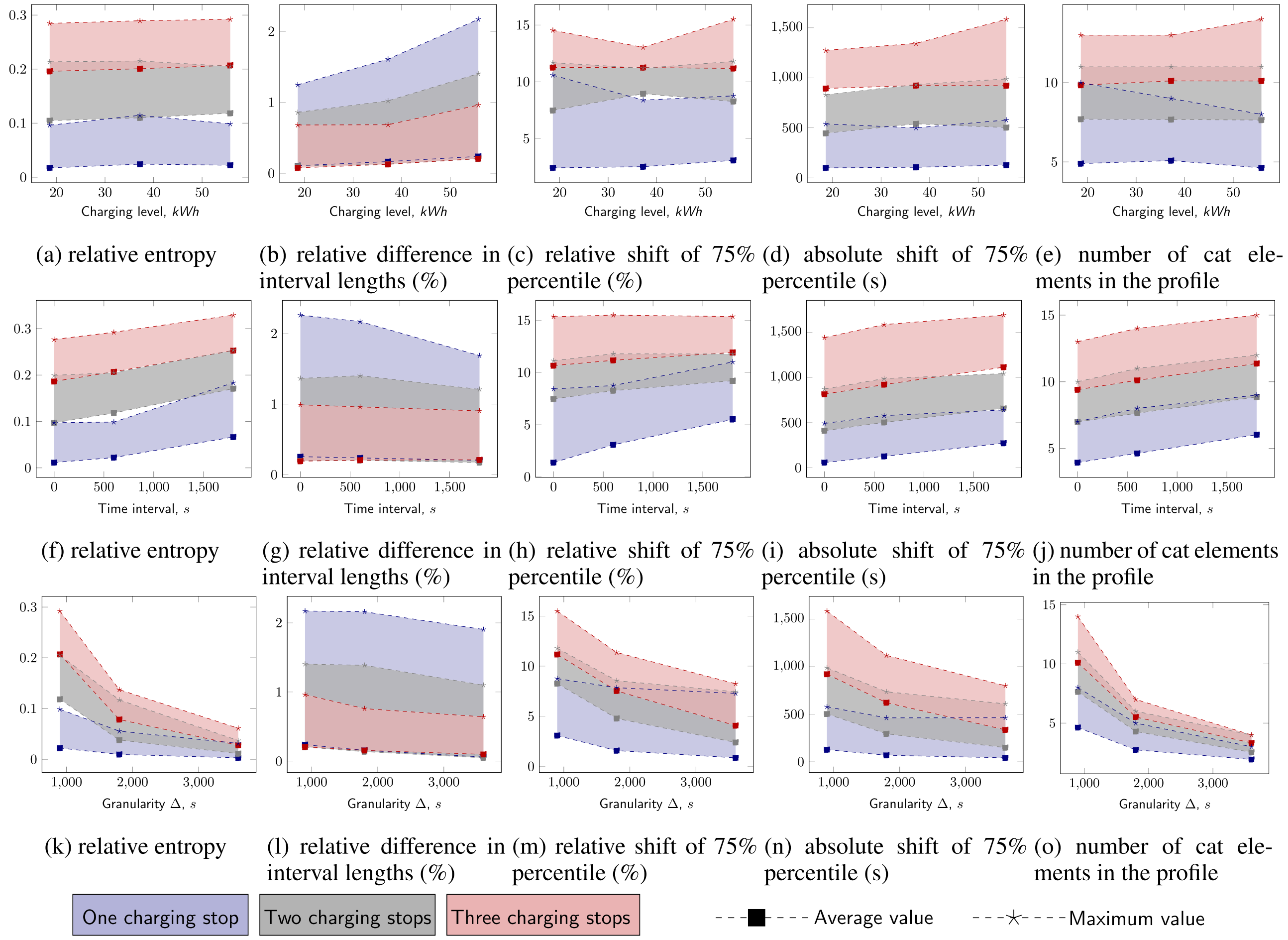
Parameter	Values
Charge level, % of a 62 kWh battery	30%, 60%, **90%**
Departure time interval, minutes	0, **10**, 30
Time granularity Δ, minutes	**15**, 30, 60

All graphs include three settings on the x-axis and the results after each of the three consecutive charging stops along the route are shown in different colors. Each result is represented by the maximum observed value of the measured metric (upper curve; star marks in the figure) and the average value (lower curve; square marks in the figure).

2) THE RESULTS

The first row in Figure 13 shows the results of the experiments with different maximum charging levels in *kWh* applied at a charging station. The second row shows the results when changing the length of the departure time interval. Note that while Definition 17 gets the departure time point as an input, it is trivial to extend it to get a time interval instead. This is so because, starting from the second leg, departure times are intervals stemming from the CAT profiles of previous legs. Finally, the third row gives the results of experiments with different time granularities. Without changing anything in the underlying simulation, the experiment adjusts Δ—the size of a CAT element in a profile. Table 2 summarizes the values used for these parameters.

The first column of the results shows that the relative entropy is significantly above zero, signifying that CAT profiles capture probability distributions significantly different from uniform. Further, the difference becomes more pronounced for routes with more legs—Figure 13a demonstrates that adding one charging stop increases the relative entropy by approximately 0.1. As expected, Figure 13k shows that increasing Δ, and thus decreasing the time granularity,



(a) relative entropy (b) relative difference in interval lengths (%) (c) relative shift of 75% percentile (%) (d) absolute shift of 75% percentile (s) (e) number of cat elements in the profile

(f) relative entropy (g) relative difference in interval lengths (%) (h) relative shift of 75% percentile (%) (i) absolute shift of 75% percentile (s) (j) number of cat elements in the profile

(k) relative entropy (l) relative difference in interval lengths (%) (m) relative shift of 75% percentile (%) (n) absolute shift of 75% percentile (s) (o) number of cat elements in the profile

FIGURE 13. Experimental results (each row of graphs is associated with the same tested parameter on the x-axis, and each column of graphs identifies the same tested parameter on the y-axis).

produces CAT profiles consisting of fewer elements and, thus, more similar to simple bounding boxes.

The experiments in the second column of the results test how CAT profiles limit the growth of uncertainty in routes with charging stops when compared to simple bounding boxes that do not capture the interdependence between travel time and energy use (cf. the end of Section II-E). The experiments show that the reduction of the length of the predicted arrival time interval is only a few percent and, interestingly, it is greatest for the routes with a single charging stop (light blue band in the graphs). This is mostly due to the fact that the long trips of the experiments use major roads outside urban areas where the variation of speed (travel time) is not big.

The third and the fourth columns of graphs show the experimental results where the difference between the modeling power of CAT profiles and simple bounding boxes is most clear. As hinted by the measurements of the relative entropy, the probability distribution captured by the CAT profiles is not uniform. That is why the 75% percentile of arrival time substantially differs from the one implied by the uniform distribution. As expected, the difference is more pronounced for multi-leg routes (compare the red band representing routes with three charging stops with the blue band for routes with a single charging stop). For example, for routes with three charging stops, the arrival time predicted by the two models differs by up to 20 minutes.

As expected, the last column of graphs shows that increasing the length of the departure time interval (Figure 13j) or decreasing Δ (Figure 13o) increases the number of CAT profile elements while changing the charging level has virtually no effect (Figure 13e).

In summary, the experiments demonstrate the modeling power of CAT profiles, especially when used to compute the likely arrival time of long-distance EV routes.

V. RELATED WORK

Data models for EV routing usually consist of a road network, traffic information, charging stations, and EV with its properties. Depending on the problem to be solved, different parts of the model are important.

A. ROAD NETWORK, TRAFFIC INFORMATION, AND ENERGY CONSUMPTION

In most recent studies on EV routing, the road network is modeled as a directed weighted graph where charging stations are a subset of vertices. Then, an edge weight is a combination of driving time and energy consumption needed to traverse the edge. Baum et al. [32] enhance the road network graph with auxiliary data: functions that model travel time and energy consumption, including minimum and maximum travel times for graph edges. Each vertex representing a charging station has a charging function that maps the charging time and the arrival state of charge to the departure state of charge. The charging function propagation algorithm uses labels of constant size, and each label stores data about all trade-offs between charging time and state of charge. Nunzio et al. [33] assign the energy required to travel along an edge as its cost. The proposed energy consumption model considers both accelerations and the impact of the road infrastructure, separating the costs of all the possible turning movements in the transportation network using the adjoint graph. In another study [34], authors generate a speed profile on each arc of the routing graph depending on the dynamic traffic data retrieved from online routing services. We assume a speed profile for each leg of the route. As we focus on modeling uncertainty, this limits the accumulation of uncertainty that would happen if speed profiles of many individual edges were concatenated into a leg of a route.

Fiori et al. [35] study how driving habits influence energy consumption. Results show that drivers who sacrifice travel time experience significant energy consumption savings due to regeneration. The minimum consumption was observed when many vehicles were assigned to the congested and low-speed arterial routes. The study applied an energy consumption model that includes instantaneous speed, acceleration, and grade information. Petkevičius et al. [36] experiment on a large dataset derived from a fleet of largely identical EVs and demonstrate how deep learning can be used to predict energy-consumption interval for a path, which is directly applicable when computing a CAT profile.

To traverse an edge of the road network might take different time depending on the current traffic situation. Hu et al. [5] assign a time-dependent, uncertain weight to each edge based on historical GPS records. They propose to maintain compact histograms to ensure their accuracy. Pedersen et al. [4] propose an algorithm to instantiate time-dependent, uncertain edge weights that satisfy FIFO property and time-dependent, uncertain contraction hierarchies to support stochastic routing. They also use histograms to represent travel cost distribution when traversing an edge. Note that CAT profiles are in effect arrival-time histograms augmented with charge intervals.

B. CHARGING STATIONS AND CHARGING FUNCTIONS

Baum et al. [37] use a realistic consumption model that includes vehicle speed, the slope of the road, and additional coefficients. Zündorf [38] uses a function that assigns a charging time to the charging stations. Piecewise linear functions are used. Montoya et al. [39] consider nonlinear charging functions and adding a charging mode to the charging station in the model. To track charging level, breakpoints of the piecewise linear approximation are used similarly to Zündorf [38]. They also propose a hybrid metaheuristic that combines iterated local search. De Cauwer et al. [40] use the model that makes a data-driven (road information, weather information, and temporal data) energy consumption prediction using energy consumption as a cost function.

C. ROUTING

Although routing algorithms are not the focus of our work, we briefly survey the main EV routing studies and the modeling assumptions made in these studies. Baum et al. [37] emphasize that route planning for electric vehicles should include path and *speed* recommendations as large speed might increase battery consumption. Battery capacity is an additional constraint, thus, the authors consider a constrained shortest path. The authors introduce the tradeoff function that maps the desired travel time to energy consumption along the graph edge. The mapped value can be negative if the vehicle recuperates. Also, such a setup introduces bounds: very slow movement can interfere traveling of other vehicles, and fast movement is very inefficient.

Rajan and Ravishankar [41] research EV routing when both travel time and energy consumption are stochastic. They generalize the Charging Function Propagation algorithm [2] for this setting. The travel time on each edge is a random variable with a known probability distribution, estimated from real traffic data. The energy consumption along the edge is a function which maps travel time to the battery energy depleted by travel. In contrast to our work, while travel time and energy consumption are stochastic, they are not time dependent.

Asghari et al. [42] stress that to find the most reliable route, the probability distribution of travel times of the route have to be computed. The authors address the problem of computing link travel time distributions to support the estimation of the travel time given a link-entrance-time. Andelmin and Bartolini [43] model the green vehicle routing problem by using a multigraph. The nodes represent customers, and arcs — trips from one customer to another. One arc defines one possible sequence of charging stations visited on the way. The path between two nodes has cardinality that is a number of fuel stations visited along the trip. They compute non-dominated refuel paths that for each pair of fuel stations on the path should include a path that has cardinality less than or equal to cardinality of the shortest path between the particular fuel stations.

Yang et al. [44] present a path-centric paradigm to estimate path cost accurately and efficiently. They enhance an edge-centric model with a path weight function that returns the joint distribution of travel costs along the edges of the path. Joint distributions of non-unit paths are derived based on distributions of unit paths (single edges). The joint distribution captures the dependency of the travel cost among edges. Multi-dimensional histograms are used to describe joint distributions.

We model a route between start and destination locations as a sequence of paths between charging stations, where each path has a CAT profile associated with it. Combining CAT profiles of paths produces a CAT profile of the whole route.

VI. CONCLUSION AND FUTURE WORK

The paper contributes with a CAT profile model to support detailed route planning for long-distance EV trips. Assuming time-dependent likely intervals of travel time and energy use for paths in a road-network as well as time-dependent waiting times and charging curves of chargers, the model captures arrival-time probability distribution and the corresponding expected arrival charge levels at the end of a route with possible charging stops.

Experimental study performed on rich semi-synthetic data demonstrates the advantages of CAT profiles when compared to a simpler bounding-box model. Specifically, CAT profiles capture inherently non-uniform arrival time probability distributions which results in more accurate predictions of probable arrival time, especially for long routes with charging stops.

As an interesting direction for future work, we envision research on filter-refinement routing algorithms, that could use CAT profiles for the detailed planning of the most promising routes, identified using simple and more cost-effective methods. Additionally, it would be interesting to explore methods for more efficient computations of CAT profiles, as the current experiments focused on exploring the properties of the model, but did not focus on efficient computation.

APPENDIX

RESTRICTED TRAVEL-TIME FUNCTION

Restricted travel-time function $\hat{tt}^{\vdash}(P, \overline{t_s}, \overline{t_a})$ computes the interval of possible travel times given an interval of departure times, $\overline{t_s} = [t_s^{\vdash}, t_s^{\dashv}]$, and an interval of arrival times, $\overline{t_a} = [t_a^{\vdash}, t_a^{\dashv}]$.

The start of the interval $\hat{tt}^{\vdash}$ is defined as follows:

$$\hat{tt}^{\vdash}(P, \overline{t_s}, \overline{t_a}) = \begin{cases} t_a^{\vdash} - t_s^{\dashv} & \text{if } \iota^{\dashv} \le t_a^{\vdash} - t_s^{\dashv} \\ \min(\iota^{\dashv}, tt') & \text{if } t_a^{\vdash} - t_s^{\dashv} < \iota^{\dashv} \le t_a^{\dashv} - t_s^{\dashv} \wedge \quad (1) \\ & \quad\ \ \iota^{\vdash} \le t_a^{\vdash} - t_s^{\vdash} \quad (2) \\ \min(\iota^{\vdash}, \iota^{\dashv}) & \text{if } t_a^{\vdash} - t_s^{\dashv} < \iota^{\dashv} \le t_a^{\dashv} - t_s^{\dashv} \wedge \\ & \quad\ \ \iota^{\vdash} > t_a^{\vdash} - t_s^{\vdash} \quad (3) \\ \min(\iota^{\vdash}, tt'') & \text{if } \iota^{\dashv} > t_a^{\dashv} - t_s^{\dashv} \wedge \iota^{\vdash} \ge t_a^{\vdash} - t_s^{\vdash} \quad (4) \\ tt' & \text{if } \iota^{\dashv} > t_a^{\dashv} - t_s^{\dashv} \wedge \iota^{\vdash} < t_a^{\vdash} - t_s^{\vdash} \quad (5) \end{cases}$$

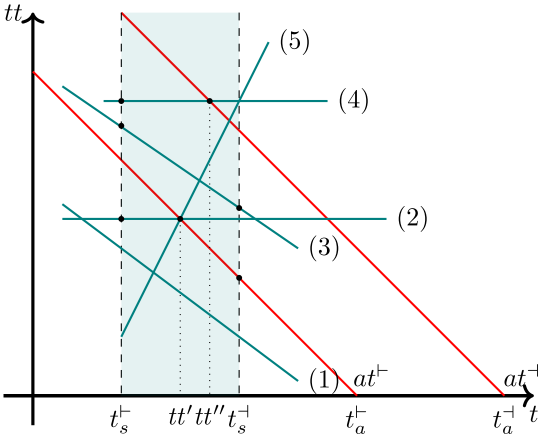


FIGURE 14. Computing the lower bound of the restricted travel-time function.

where

$$\iota^{\vdash} = tt^{\vdash}(P, t_s^{\vdash}), \iota^{\dashv} = tt^{\vdash}(P, t_s^{\dashv})$$

$$tt' = xtion(at^{\vdash}, tt^{\vdash}(P, \cdot)), tt'' = xtion(at^{\dashv}, tt^{\vdash}(P, \cdot))$$

$$xtion(t_a, bf(\cdot)) = \frac{\Delta\iota(t_a - t_s^{\vdash}) + \Delta t_s \cdot \iota^{\vdash}}{\Delta t_s + \Delta\iota}$$

$$\Delta\iota = \iota^{\dashv} - \iota^{\vdash}, \Delta t_s = t_s^{\dashv} - t_s^{\vdash}$$

The end of the interval $\hat{tt}^{\dashv}$ is a mirror case and can be defined in the same manner.

Figure 14 illustrates all five cases of Definition 12. Blue lines show the different cases for the lower bound of the travel-time function. Black points identify points to consider, when calculating the lower bound of restricted travel-time function in cases marked with numbers. In all cases except case (1), the points are on the line. The restricted travel-time function value of case (1) is represented as a point on line $at^{\vdash}$ at $t = t_s^{\dashv}$. In cases (2) and (4), the minimal value should be selected out of both identified points on line (2) and (4), respectively, and both values are equal in the figure (see *tt-axis*). Case (3) considers the points where the line intersects $t = t_s^{\vdash}$ and $t = t_s^{\dashv}$. Finally, case (5) means that the minimal value of the travel function is at the point where the line intersects with the lower bound of the arrival time interval.

REFERENCES

[1] G. Erbach, "Co2 emission standards for new cars and vans," Eur. Parliamentary Res. Service, Brussels, Belgium, Briefing PE 698.920, May 2023. [Online]. Available: https://www.europarl.europa.eu/RegData/etudes/BRIE/2022/698920/EPRS_BRI(2022)698920_EN.pdf

[2] M. Baum, J. Dibbelt, A. Gemsa, D. Wagner, and T. Zündorf, "Shortest feasible paths with charging stops for battery electric vehicles," 2019, *arXiv:1910.09812*.

[3] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. F. Werneck, "Route planning in transportation networks," in *Algorithm Engineering: Selected Results and Surveys*, L. Kliemann and P. Sanders, Eds. Cham, Switzerland: Springer, 2016, pp. 19–80.

[4] S. A. Pedersen, B. Yang, and C. S. Jensen, "Fast stochastic routing under time-varying uncertainty," *VLDB J.*, vol. 29, no. 4, pp. 819–839, Jul. 2020.

[5] J. Hu, B. Yang, C. S. Jensen, and Y. Ma, "Enabling time-dependent uncertain eco-weights for road networks," *GeoInformatica*, vol. 21, no. 1, pp. 57–88, Jan. 2017.

[6] A Better RoutePlanner. (2018). *Tesla Model 3 Performance vs RWD Consumption—Real Driving Data From 233 Cars*. [Online]. Available: https://forum.abetterrouteplanner.com/blogs/entry/22-tesla-model-3-performance-vs-rwd-consumption-real-driving-data-from-233-cars/

[7] Tesla Motors Club. (2019). *New Superchargers V3*. [Online]. Available: https://teslamotorsclub.com/tmc/threads/new-superchargers-v3.144597/

[8] A. Barauskas, A. Brilingaitė, L. Bukauskas, V. Čeikutė, A. Čivilis, and S. Šaltenis, "Test-data generation and integration for long-distance e-vehicle routing," *GeoInformatica*, vol. 27, no. 4, pp. 737–758, Oct. 2023.

[9] K. Sung, M. G. H. Bell, M. Seong, and S. Park, "Shortest paths in a network with time-dependent flow speeds," *Eur. J. Oper. Res.*, vol. 121, no. 1, pp. 32–39, Feb. 2000.

[10] E. Kanoulas, Y. Du, T. Xia, and D. Zhang, "Finding fastest paths on a road network with speed patterns," in *Proc. 22nd Int. Conf. Data Eng. (ICDE)*. Atlanta, GA, USA: IEEE, 2006, pp. 1–10.

[11] M. Setak, M. Habibi, H. Karimi, and M. Abedzadeh, "A time-dependent vehicle routing problem in multigraph with FIFO property," *J. Manuf. Syst.*, vol. 35, pp. 37–45, Apr. 2015.

[12] R. Jaballah, R. Ramalho, J. Renaud, and L. C. Coelho, "The impact of time aggregation and travel time models on time-dependent routing solutions," *INFOR, Inf. Syst. Oper. Res.*, vol. 61, no. 4, pp. 490–508, Nov. 2023.

[13] M. P. Wellman, M. Ford, and K. Larson, "Path planning under time-dependent uncertainty," in *Proc. 11th Conf. Uncertainty Artif. Intell.* San Francisco, CA, USA: Morgan Kaufmann Publishers, 1995, pp. 532–539.

[14] M. P. Wellman, M. Ford, and K. Larson, "Path planning under time-dependent uncertainty," 2013, *arXiv:1302.4987*.

[15] B. B. Krogh, O. Andersen, and K. Torp, "Analyzing electric vehicle energy consumption using very large data sets," in *Database Systems for Advanced Applications*. Hanoi, Vietnam: Springer, Apr. 2015, pp. 471–487.

[16] Institut fuer Theroretische Informatik and Karlsruher Institut fuer Technology (KIT). *KaTCH—Karlsruhe Time-Dependent Contraction Hierarchies*. Accessed: Mar. 16, 2021. [Online]. Available: https://github.com/GVeitBatz/KaTCH

[17] G. V. Batz, R. Geisberger, P. Sanders, and C. Vetter, "Minimum time-dependent travel times with contraction hierarchies," *ACM J. Experim. Algorithmics*, vol. 18, pp. 1.1–1.43, Dec. 2013, Art. no. 1.4, doi: 10.1145/2444016.2444020.

[18] N. Makulavicius, A. Brilingaite, L. Bukauskas, A. Civilis, V. Krinickij, and S. Saltenis, "E-tri: E-vehicle testbed routing infrastructure," in *Proc. 31st ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.* Hamburg, Germany: ACM, 2023, p. 4.

[19] OpenStreetMap Foundation. (2023). *Openstreetmap*. [Online]. Available: https://www.openstreetmap.org

[20] A. Jarvis, H. I. Reuter, A. Nelson, and E. Guevara, "Hole-filled seamless SRTM data V4," Int. Centre Tropical Agricult. (CIAT), 2008. [Online]. Available: https://srtm.csi.cgiar.org

[21] *Srtm 90m DEM Digital Elevation Database*, CGIAR–Consortium Spatial Inf. (CGIAR-CSI), Int. Centre Tropical Agricult. (CIAT), Cali-Palmira, 2018. [Online]. Available: https://csidotinfo.wordpress.com/data/srtm-90m-digital-elevation-database-v4-1/

[22] *Open Charge Map—The Global Public Registry of Electric Vehicle Charging Locations*, Supported by a Community of Businesses, Charities, Developers and Interested Parties, Open Charge Map, 2023. [Online]. Available: https://openchargemap.org/site/

[23] *Sumo—Simulation of Urban Mobility*, DLR, German Aerospace Center (DLR) and Others, Cologne (Köln), Germany, 2023. [Online]. Available: https://sumo.dlr.de/docs/

[24] C. Gawron, "An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model," *Int. J. Modern Phys. C*, vol. 9, no. 3, pp. 393–407, May 1998.

[25] L. Codeca, R. Frank, S. Faye, and T. Engel, "Luxembourg SUMO traffic (LuST) scenario: Traffic demand evaluation," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 2, pp. 52–63, May 2017.

[26] Z. Shou, X. Chen, Y. Fu, and X. Di, "Multi-agent reinforcement learning for Markov routing games: A new modeling paradigm for dynamic traffic assignment," *Transp. Res. C, Emerg. Technol.*, vol. 137, Apr. 2022, Art. no. 103560.

[27] TomTom International BV. (2020). *Tomtom Traffic Index. Berlin Traffic Report*. [Online]. Available: https://www.tomtom.com/traffic-index/berlin-traffic/

[28] Google Maps Platform. *Routes & Directions*. Google Cloud, Accessed: May 30, 2023. [Online]. Available: https://cloud.google.com/maps-platform/routes/

[29] T. Kurczveil, P. Á. López, and E. Schnieder, "Implementation of an energy model and a charging infrastructure in sumo," in *Proc. Simul. Urban MObility User Conf.* Berlin, Germany: Springer, 2013, pp. 33–43.

[30] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582. [Online]. Available: https://elib.dlr.de/127994/

[31] M. S. Islam and N. Mithulananthan, "Daily EV load profile of an EV charging station at business premises," in *Proc. IEEE Innov. Smart Grid Technol. Asia (ISGT-Asia)*. Melbourne, VIC, Australia, Nov. 2016, pp. 787–792.

[32] M. Baum, J. Dibbelt, A. Gemsa, and D. Wagner, "Towards route planning algorithms for electric vehicles with realistic constraints," *Comput. Sci. Res. Develop.*, vol. 31, nos. 1–2, pp. 105–109, May 2016.

[33] G. De Nunzio, L. Thibault, and A. Sciarretta, "A model-based eco-routing strategy for electric vehicles in large urban networks," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Rio de Janeiro, Brazil, Nov. 2016, pp. 2301–2306.

[34] G. De Nunzio, I. Ben Gharbia, and A. Sciarretta, "A time- and energy-optimal routing strategy for electric vehicles with charging constraints," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Rhodes, Greece, Sep. 2020, pp. 1–8.

[35] C. Fiori, K. Ahn, and H. A. Rakha, "Optimum routing of battery electric vehicles: Insights using empirical data and microsimulation," *Transp. Res. D, Transp. Environ.*, vol. 64, pp. 262–272, 2018.

[36] L. Petkevicius, S. Saltenis, A. Civilis, and K. Torp, "Probabilistic deep learning for electric-vehicle energy-use prediction," in *Proc. 17th Int. Symp. Spatial Temporal Databases*, Aug. 2021, pp. 85–95.

[37] M. Baum, J. Dibbelt, D. Wagner, and T. Zündorf, "Modeling and engineering constrained shortest path algorithms for battery electric vehicles," in *Proc. 25th Annu. Eur. Symp. Algorithms (ESA)*. Vienna, Austria: Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2017, p. 11.

[38] T. Zündorf, "Electric vehicle routing with realistic recharging models," Master's thesis, Karlsruhe Inst. Technol., Karlsruhe, Germany, 2014. [Online]. Available: https://i11www.iti.kit.edu/_media/teaching/theses/ma-zuendorf-14.pdf

[39] A. Montoya, C. Guéret, J. E. Mendoza, and J. G. Villegas, "The electric vehicle routing problem with nonlinear charging function," *Transp. Res. B, Methodol.*, vol. 103, pp. 87–110, Sep. 2017.

[40] C. De Cauwer, W. Verbeke, J. Van Mierlo, and T. Coosemans, "A model for range estimation and energy-efficient routing of electric vehicles in real-world conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 2787–2800, Jul. 2020.

[41] P. Rajan and C. V. Ravishankar, "Stochastic route planning for electric vehicles," in *Proc. 20th Int. Symp. Experim. Algorithms (SEA)*, C. Schulz and B. Uçar, Eds., vol. 233. Heidelberg, Germany: Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022, p. 15.

[42] M. Asghari, T. Emrich, U. Demiryurek, and C. Shahabi, "Probabilistic estimation of link travel times in dynamic road networks," in *Proc. 23rd SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.* Bellevue, WA, USA: ACM, 2015, pp. 1–10.

[43] J. Andelmin and E. Bartolini, "An exact algorithm for the green vehicle routing problem," *Transp. Sci.*, vol. 51, no. 4, pp. 1288–1303, Nov. 2017.

[44] B. Yang, J. Dai, C. Guo, C. S. Jensen, and J. Hu, "PACE: A PAth-CEntric paradigm for stochastic path finding," *VLDB J.*, vol. 27, no. 2, pp. 153–178, Apr. 2018.

SIMONAS ŠALTENIS received the Ph.D. degree in computer science from Aalborg University, Denmark. He is a Research Professor with the Institute of Computer Science, Vilnius University; and an Associate Professor with the Department of Computer Science, Aalborg University. His research interests focus on spatial and spatio–temporal data management and intelligent transportation systems.

ANDRIUS BARAUSKAS received the Ph.D. degree in technological science from Vilnius Tech, Lithuania. He is a Project Researcher with the Institute of Computer Science, Vilnius University. His research interests focus on transport modeling and spatial planning.

AGNĖ BRILINGAITĖ received the Ph.D. degree in computer science from Aalborg University, Denmark. She is an Associate Professor with the Institute of Computer Science, Vilnius University. Her research interests focus on spatial data modeling, location-based services, cybersecurity training, and education in computer science.

LINAS BUKAUSKAS (Member, IEEE) received the Ph.D. degree in computer science from Aalborg University, Denmark. He is an Associate Professor with the Institute of Computer Science, Vilnius University. His research interests include cybersecurity, data processing, database management systems, data mining, and natural language processing.

VAIDA ČEIKUTĖ received the Ph.D. degree in computer science from Aarhus University, Denmark. She is a Project Researcher with the Institute of Computer Science, Vilnius University. Her research interests include trajectory pattern mining, geo-context in location-based services, and intelligent transportation systems.

ALMINAS ČIVILIS received the Ph.D. degree in computer science from Vilnius University, Lithuania. He is an Assistant Professor and a Project Researcher with the Institute of Computer Science, Vilnius University. His research interests focus on intelligent transportation systems and location-based services.