Reducing Waiting Times at Charging Stations with Adaptive Electric Vehicle Route Planning

Sven Schoenberg, *Student Member, IEEE,* Falko Dressler, *Fellow, IEEE*

***Abstract*—Electric vehicles are becoming more popular all over the world. With increasing battery capacities and a growing fast-charging infrastructure, they are becoming suitable for long-distance travel. However, queues at charging stations could lead to long waiting times, making efficient route planning even more important. In general, optimal multi-objective route planning is extremely computationally expensive. We propose an adaptive charging and routing strategy, which considers driving, waiting, and charging time. For this, we developed a multi-criterion shortest-path search algorithm using contraction hierarchies. To further reduce the computational effort, we precompute shortest-path trees between the known locations of the charging stations. We propose a central charging station database (CSDB) that helps estimating waiting times at charging stations ahead of time. This enables our adaptive charging and routing strategy to reduce these waiting times. In an extensive set of simulation experiments, we demonstrate the advantages of our concept, which reduces average waiting times at charging stations by up to 97 %. Even if only a subset of the cars uses the CSDB approach, we can substantially reduce waiting times and thereby the total travel time of electric vehicles.**

I. Introduction

THE capacity of batteries used by electric vehicles is continuously increasing and the trend is projected to continue for the coming years [1]. Together with a growing fast-charging infrastructure, this makes electric vehicles more suitable for long-distance travel. Nevertheless, if recharging on a trip is necessary, the charging times can still be quite long and as more electric vehicles utilize the charging infrastructure, long waiting times are becoming a main challenge [2]. There is a substantial heterogeneity in the charging infrastructure, for example with respect to available charging power, location, and potential waiting time. Thus, the total travel time of the trip can be significantly impacted by the selection of charging stations along the trip. We therefore need solutions helping to plan long-distance trips and coordinating the use of charging stations between vehicles.

In general, it has been discovered that, due to the need to take recharging into account, route planning for electric vehicles is more challenging compared to route planning for conventional vehicles [3]–[5]. The recharging time of a vehicle depends on multiple factors. Most importantly, these are the amount to recharge, which depends on the energy consumed while driving, and the power of the charging station. The driver has the option to take faster routes or more economic routes, i.e. routes that require less energy. In order to minimize the total travel time, the driver could take a more economic route to save charging time – at the cost of a longer driving time. The total travel time is also influenced by the concrete amount of energy to be charged at the charging station. This is particularly complicating the optimization problem as the charging process is not linear: Lithium-ion batteries, as used in most electric vehicles, are charged with a charging protocol that decreases the charging speed considerably after the battery capacity reaches about 80 % [6].

Waiting times due to fully occupied charging stations can be a substantial part of the total travel time as well [7]. If the potential waiting time at each charging station is known ahead of time, it can be taken into account when planning the route. It might take less time to take a detour to a different charging station, if the waiting time there is substantially smaller. We also think it would make sense to take a slower, more energy efficient route, if it is known that the car has to wait at the charging station anyhow.

To solve the problem of finding a route that minimizes the total travel time for an electric vehicle, we have to select the charging stations, the amount of energy to charge at each charging station, and a route to get to and from the charging stations. The charging station selection must take into account the driving time, the potential waiting time and the charging time. The driving time, as well as the charging time, depends on the selected route to the charging station. These aspects also mutually influence each other.

Our solution is to use a multi-criterion shortest-path search for the criteria energy and time, revealing all Pareto-optimal routes from the most economic to the fastest route. In a second step, the alternatives are compared to identify the best candidate. The key challenge is that such a multi-criterion shortest-path search is rather expensive with respect to required computational resources [8]. In our previous work [9], we presented an approach to accelerate the search by focusing on the most popular queries, which are between the (known) locations of the charging stations. We selected the charge amount and the route between charging stations with an adaptive charging and routing strategy, to minimize the total travel time.

In this paper, our main contributions can be summarized as follows:

- we present a way for electric vehicles to coordinate their (planned) charging station visits to reduce the waiting

Research reported in this paper was conducted in part in the context of the Hy-Nets4all project, supported by the European Regional Development Fund (EFRE-0801678).

Sven Schoenberg is with the Software Innovation Lab and the Department of Computer Science, Paderborn University, Germany. E-Mail: sven.schoenberg@c-lab.de

Falko Dressler is with the School of Electrical Engineering and Computer Science, TU Berlin, Germany. E-Mail: dressler@ccs-labs.org

time. We propose a central charging station database (CSDB), to which the vehicles announce their planned charge stops and which, in return, can estimate waiting times at charging stations. The estimated waiting times are used by our existing charging station routing algorithm to minimize the total travel time, extending our prior work in [9]. Using the database is not mandatory, but it provides significant benefits to its users even if only a small subset of all vehicles uses it; and
- we perform a simulation study to analyze the effect of different penetration rates of the CSDB and of using historical data on the travel times to study our adaptive charging and routing strategy compared to related strategies. We show that by using the CSDB we can reduce the average waiting time up to 97 %.

The rest of the paper is organized as follows. Related works are discussed in Section II. In Section III, we describe our charging station routing approach, introduced in [9]. In Section IV we present the coordination of electric vehicle charging with the CSDB and in Section V we do an extensive performance evaluation. Lastly, Section VI concludes with a summary and future work.

II. Related Work

The classic route planning in a road traffic network is a shortest path problem, where the best route to be found from A to B is based on some criterion. Typically, the criterion to optimize for is either driving distance (shortest route), travel time (fastest route), energy consumption (economic route), or a combination thereof. The best known solution is Dijkstra's algorithm [10]. Depending on the size of the graph, Dijkstra's algorithm can be too slow in practice, but several techniques have been proposed to speed things up. For example, the A* algorithm [11] uses a heuristic for a directed search. If the heuristic is guaranteed not to overestimate the cost, A* will find the optimal path.

Another technique is the use of contraction hierarchies introduced by Geisberger et al. [12]. In a preprocessing step, shortcuts are added to the graph that can later speed-up the path finding query significantly. This is done by contracting the nodes of the graph one by one. Each node that is contracted, is effectively removed from the graph. If the node was part of the shortest path between two of its neighbors, a direct edge between these neighbors (shortcut) is added to ensure that the shortest path is maintained. Whether this is the case can be determined by doing a shortest-path search with Dijkstra's algorithm from each neighbor to all other neighbors. Each node is assigned a level based on the order of contraction. A higher level indicates that the node was contracted later and its shortcuts might have replaced shortcuts of lower level nodes. To query the shortest path, a bidirectional search with Dijkstra's algorithm is done with both sides only traversing to nodes that have a higher level until they meet. This way, the number of nodes that need to be visited to find the shortest path is reduced significantly. To further speed-up the query, A* can be used instead of Dijkstra's algorithm for the bidirectional search [13].

Route planning for electric cars presents additional challenges. The constraints of the battery, especially the limited range, have to be accounted for. This can further include recuperation, also called regenerative braking, i.e., charging the battery when slowing down or driving downhill. Finding the shortest path that also considers such battery constraints is a constrained shortest path (CSP) problem [14]. To find the fastest route that is reachable with a limited range, a multi-criterion shortest-path search can be performed using the criteria travel time and energy consumption. This results in all Pareto optimal paths for these criteria and we can, for example, choose the one with the best travel time that still fulfills the energy constraints. To calculate all Pareto optimal paths, a modified version of either Dijkstra's algorithm or the A* algorithm can be used; however, due to the increased complexity, this is even less practical than for a single criterion. Fortunately, contractions hierarchies can also be used to speed-up multi-criteria path finding and to solve the CSP in acceptable time [15].

The preprocessing step for contractions hierarchies for multi-criteria path finding requires significant computational effort for large (e.g., country-sized) maps. As more and more nodes are contracted, the remaining uncontracted nodes get more and more neighbors, which makes contracting the last few nodes very expensive. It is possible to restrict the preprocessing to only contract a subset of all nodes. For example, Storandt [16] contracted only 99.5 % of the nodes to achieve reasonable preprocessing times. The remaining uncontracted graph is called a core graph [17]. This can substantially save preprocessing time, but also causes higher query times [9].

Another approach by Baum et al. [13] introduces a way to solve the electric vehicle continuous adaptive speeds shortest path (EVCAS) problem, which they define as finding a feasible path that respects battery constraints, considers variable speed and minimizes driving time. To speed-up queries, they use contraction hierarchies together with A* and can compute optimal solutions in less than a second even for large battery sizes and country sized maps. For interactive applications, very low query times are more important than getting optimal solutions. By using a heuristic, query times can be improved significantly at the cost of some inaccuracy [13], [18].

The problem becomes more complicated when recharging on the way is also considered, to enable long-distance trips. One solution is to limit the number of recharging events and choosing the most economic route that is not more than 10 % longer than the shortest route [15]. Morlock et al. [19] solve the problem by first querying a number of potential routes ranging from shortest to fastest from a conventional routing service and treating the routes as a reduced graph. They then integrated the charging stations into the graph and performed a multi-objective shortest-path search to find the fastest route from start to destination. Most publications assume a full recharge at each charging station [7], [8], [15], [19], [20], only few also consider partial charging [3], [5].

One assumption in these works is that they consider only one vehicle and assume it can immediately start charging when it arrives at a charging station. In reality, there will often be multiple vehicles that want to charge at the same time, but the

number of vehicles that can charge concurrently is limited by the number of charge points of the charging station. If there is no coordination between the vehicles, this will lead to queues and long waiting times. A possible solution to this problem is a reservation system, where the vehicles can reserve a time slot at a charging station in the future and can therefore plan their trip accordingly to avoid waiting times. Many publications make use of such a reservation system [21]–[26], sometimes with the possibility to update the reservation, if needed [22]. Most of these systems have a first-come-first-serve policy, but some can also prioritize reservations, leading to cases where high-priority vehicles can charge before others even if they arrive later at the charging station [24]. Hou et al. [25] use a scheduler that allocates reservations based on user given information about their time preferences. Due to the assumption that users are selfish and do not want to reveal their true time preferences to avoid unfavorable time slots, they propose an iterative auction which, by progressively eliciting the users' preferences as necessary, preserves their privacy.

Recently, some approaches for charging station and route selection have been using deep reinforcement learning. It enables them to make complex decisions in a stochastic environment with changing conditions like traffic, weather, dynamic charge prices etc. by learning an optimal policy. Qian et al. [27] present a charging navigation solution which aims to minimize the total travel time and charging cost. It can take into account waiting times at charging stations, traffic conditions and charge prices, thereby coordinating smart grid and intelligent transportation systems. However, they do not consider direct coordination between vehicles, but simply assume that charging stations know how long the waiting times will be. Lee et al. [26] propose a similar system where there is coordination between vehicles with a reservation system and charging decisions are made by a central service. However, both solutions suffer from poor scalability. They evaluated very small instances with graphs of only 39 nodes and three charging stations. Zhang et al. [28] use deep reinforcement learning for planning charging scheduling at a larger scale. They evaluated instances of a big city with more than 1000 charging stations. However, they use Dijkstra's algorithm for route selection and a simple energy consumption model, which only depends on the driven distance. While this may be sufficient for inner-city navigation, more sophisticated models are needed for long-distance navigation.

A different approach is a centralized service that knows about the current charging station utilization and can give vehicles advice on where to charge [29]. The vehicles could also announce their charging intentions to this service, so that it can predict the waiting time in the future. De Weerdt et al. [7] call this intention-aware routing. They combined the information about charging intentions with historical data and were able to reduce waiting times in some cases by about 80 %. However, they do not consider long trips with multiple charge stops or partial charging and assume charge stops always take a fixed time.

We go one step further and combine adaptive charging and routing strategies for long trips with a centralized service that can estimate waiting times at charging stations with the current utilization, planned charge stops of other vehicles, and historical data. In contrast to a reservation system, it does not require cooperation from charging station providers and is not mandatory for the vehicles to use. In addition, we also make use of a realistic nonlinear charging model to consider partial charging of the batteries.

III. Charging Station Routing

This section describes our initial approach [9] for the ease of the reader to understand our novel concept presented in Section IV.

The goal of our charging station routing approach is to minimize the total travel time of an electric vehicle on a long-distance trip. This includes finding a route from the origin to the destination and selecting charging stations along the way to recharge the vehicle if necessary. The total travel time consists of the driving time as well as the waiting and charging time at the charging stations. The selection of the charging stations has a big influence on the route, because the consumed energy while driving has to be recharged. Depending on the charge power of the charging station, this can lead to significant charging time. Choosing a slower but more energy-efficient route might be faster overall if it reduces the charging time at the charging station enough. When finding the route, we therefore can not simply optimize for the criterion time, i.e. take the fastest route. Instead, we find all multi-criterion shortest paths for the criteria time and energy, i.e. all routes from the fastest to the most energy efficient. We then select the route that minimizes the total travel time in combination with charge time and waiting time.

Taking all this into account, our approach consists of two parts. The first part is an approach for multi-criterion shortest-path finding that accelerates the search between the known locations of the charging stations. It enables us to query all Pareto-optimal routes from the fastest to the most energy-efficient route.

The second part is the selection of charging stations and routes that minimize the total travel time. We compare the route alternatives, returned by the multi-criterion shortest-path finding, between charging stations and the origin and destination. Our adaptive charging and routing strategy then selects the optimal combination of charging stations and routes.

A. Multi-Criterion Shortest-Path Finding

Dijkstra's algorithm can be modified to be used for multi-criterion shortest-path finding. Instead of setting a single label per node to denote the predecessor node and the minimum cost to this node, we maintain a Pareto set of labels at each node. A label contains the costs for all criteria and the predecessor node. The path finding query is a lot more computationally expensive than for a single-criterion shortest-path search. The nodes have to be visited multiple times and new labels have to be checked with existing labels for dominance to maintain the Pareto set.

We use contraction hierarchies to accelerate the path finding. Even though this improves the query times significantly, they might still be in the order of seconds or even minutes, especially for long distances of $>$200 km on a dense street network. Using

only this, the total computation time of our algorithm would be unacceptable, because it makes many queries between the origin, destination and the charging stations.

B. Shortest-Path Tree Precomputing

Most of the query time is used to explore the graph and maintain the Pareto sets of labels at the nodes. Our algorithm queries only the routes between the charging stations and the origin and destination. Because the charging station locations are always the same, we would explore the graph from the same nodes over and over again. We can avoid this, by exploring the graph from all locations of the charging stations once in a preprocessing step. The result of an exploration is a shortest-path tree, which contains the sets of labels of all explored nodes. The exploration can be limited to an energy consumption equal to the maximum battery capacity of the vehicles. Because the query of contraction hierarchies is bidirectional, we have to create the shortest-path trees twice for each charging station: Once exploring the graph forwards, and once exploring backwards. Precomputing the shortest-path trees for all charging stations is only feasible because the contraction hierarchies significantly reduce the number of nodes that have to be explored.

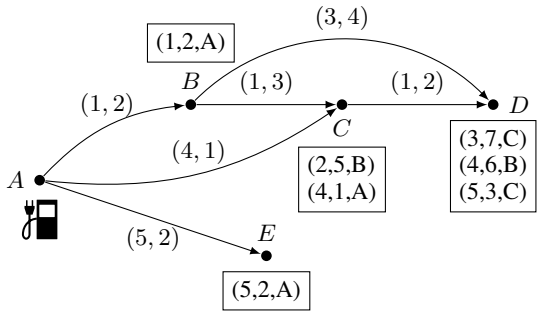
To perform a query with the shortest-path trees, we use the forward explored tree of the origin node and backward explored tree of the destination node. We then identify the nodes that are covered by both trees, i.e., nodes with Pareto sets of labels in both trees. We create the sumset of both Pareto sets for each of these nodes and remove all dominated elements. Each set contains the costs of all shortest-paths from origin to destination via that node. To get the costs of all Pareto optimal shortest-paths from origin to destination, we combine the sets of all nodes and again remove all dominated elements.

An example of such a query with shortest-path trees for two criteria is depicted in Figure 1. Each label contains the costs for both criteria and the predecessor node, which is needed to reconstruct the path later. The common nodes, which are covered by both trees, are identified and the sumsets for these nodes are created. The elements of the sumset contain the sum of the costs, as well as the predecessor nodes of both labels. Then, the elements of all sumsets are combined into one Pareto set, with each element also storing the node of the sumset. This set contains the costs of all Pareto optimal paths and the information necessary to reconstruct them.

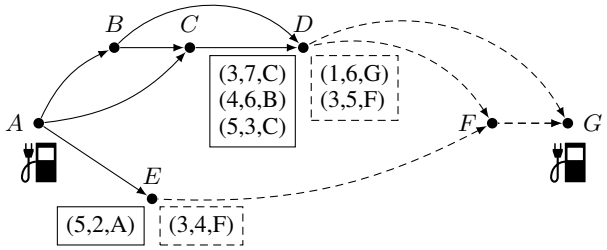
Our charging station routing algorithm makes queries between the charging stations and the origin and destination. With the shortest-path trees precomputed for all charging stations, we just need to create the shortest-path trees for the origin and destination and can then answer all queries without exploring the graph again. We do not have to reconstruct all paths; in order to save time, we can select a path based on the costs and only reconstruct the selected one.

C. Adaptive Charging and Routing Strategy

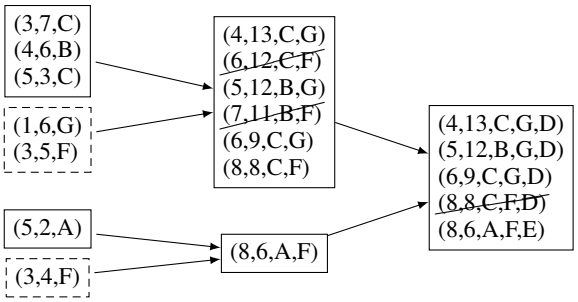
Our adaptive charging and routing strategy tries to find a route from the origin to the destination that minimizes the total travel time. If necessary, it selects charging stations along



(a) Shortest-path tree with Pareto sets of labels



(b) Forward and backward shortest-path trees with labels of common nodes. Solid: Forward tree from node A. Dashed: Backward tree from node G



(c) Create sumsets from node labels and combining them to get resulting Pareto set

Figure 1. Example query with precomputed shortest-path trees

the way and determines the amount of energy to recharge. The selection of the charging stations is done with a shortest-path search on a dynamically generated graph that connects origin, destination and charging station nodes that are reachable within the vehicle's range. We use an A* search to perform the shortest-path search on the graph. The edge weights are generated by querying all Pareto optimal paths between the two nodes and selecting the one with the lowest sum of charging time, driving time and waiting time. The heuristic is not the linear distance to the destination, but determined with a single criterion shortest-path search.

The charging time is calculated by our charging model (cf. Section V-C) and depends on the power of the charging station as well as the amount of energy to recharge. Many publications assume a full charge at every charge stop. Our adaptive charging strategy selects the amount to recharge based our charging model and the maximum charge power of the next charging station. We continue to charge as long as the maximum charge power of the next charging station is below

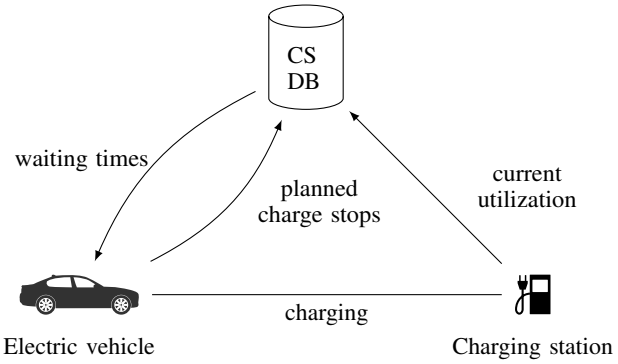


Figure 2. Charging station database concept

the current charge power. We, of course, always charge at least the energy required to be able to reach the next charging station.

In our dynamically generated graph, each edge weight contains the charging time at the current charging station, the driving time to the next charging station, and also the expected waiting time and charging time at the next charging station. We include the charging time at the next charging station, so that the route selection is influenced by how much time it takes to recharge the spent energy. Because we do not know at this point how much we will charge at the next charging station, we temporarily assume to fully recharge. The time for the full recharge will be replaced by the actual charging time that is determined when calculating the next edge weight. While exploring the graph, we propagate the batteries state of charge (SOC).

Shortest-path tree precomputing reduces the query time significantly, but it is still orders of magnitudes slower than a single-criterion shortest-path search. To keep the number of multi-criterion queries as low as possible, when we explore a node and have to calculate the edge weights to all neighbors, we set it to a temporary heuristic value based on two single-criterion shortest-path searches for the fastest and most energy-efficient routes. Only when the edge is about to be travelled as part of the A* graph exploration, because the corresponding node is at the top of the open list, do we replace the value with the accurate value from a multi-criterion shortest-path search. It might happen that the node is then no longer at the top of the open list, in which case the process has to be repeated.

IV. Charging Station Database (CSDB)

To coordinate charging station visits between the electric vehicles, we propose the use of a centralized CSDB (please note that our algorithm can also be executed using mobile edge computing [30]). It can estimate waiting times at charging stations in the future, which can be used by the vehicles when they calculate their route. To be able to do this, the vehicles have to announce their planned charge stops to the database. It also gets information about the current utilization of the charging stations and stores statistical data about past utilization. The principle of our CSDB approach is depicted in Figure 2.

With the adaptive charging and routing strategy, the charging station routing algorithm can easily take advantage of the waiting time information as an additional time cost at each charge stop. For instance, if a vehicle will have to wait at a charging station anyway, it might as well drive a slower but more energy-efficient route. Even though it arrives later, it will start charging at the same time and the reduced energy consumption saves charging time, improving the overall total travel time. Another example might be to make a detour to a different charging station if the saved waiting time is greater than the additional driving time.

Table I
Description of symbols

Symbol	Description
S	Set of charging stations
C_s	Set of charge points of charging station s
t_0	Query time
t^q_{arr}	Queried arrival time
t^q_{wait}	Resulting waiting time of query
t^q_{start}	Resulting charge start time of query
P_s	Set of planned charge stops of charging station s
t^p_{arr}	Arrival time of planned charge stop p
t^p_{start}	Charge start time of planned charge stop p
t^p_{chr}	Charging time of planned charge stop p
t^p_{dep}	Departure time of planned charge stop p
c_p	Charge point assigned to planned charge stop p
T	Period of charge stops for statistical utilization
t_{chr}	Charging time of charge stops for statistical utilization
u	Statistical utilization of charging station
n	Number of charge points of charging station

Compared to a reservation system, it does not require cooperation with charging station providers. Only information about the current utilization of the charging stations is needed, which many providers already provide as a service to potential customers. The system also does not require every vehicle to take part in it to be useful.

Apart from being able to be used with charging stations from all charging station providers, our system provides potential advantages to long-distance trip planning. When planning long-distance trips, charge stops might be planned hours in advance and the arrival time at the charging stations might deviate from the originally planned time, due to, e.g., traffic congestions. Deviations from the planned arrival time could be a problem for a reservation system. If a vehicle is late, depending on how the reservation system is implemented, the reserved charge point might be blocked for other vehicles even though it is unused, which could be a frustrating experience for other drivers. Additionally, there might not be enough time left to finish charging, if the following time slots are also reserved. Our system, on the other hand, is only using information about the vehicles' charging intentions, and assumes that the vehicles are charged on a first-come, first-served basis. This makes it potentially more flexible, because it does not matter if a vehicle is a few minutes early or late. It also means that we can simply change our planned charge stops on the fly, and that the charging stations can easily be used by vehicles not taking part in the system.

The main service that the CSDB provides is estimating

waiting times at charging stations for an arrival time in the future. The estimate is created by combining data about the current utilization of the charging station, planned charge stops of other vehicles, and the statistical utilization of the charging station. Vehicles can query estimates for any charging station and any point of time in the future. To ease the reading of the following part, we provide a list of the used symbols in Table I.

The current utilization of a charging station is known to the database in the form of occupied charge points and the time when the vehicles occupying the charge points will depart. For each charge point $c \in C_s$ of a charging station $s \in S$, we denote the departure time of the occupying vehicle as t^c_{dep}. In case the charge point is not occupied by a vehicle, we define $t^c_{dep} = t_0$, with t_0 being the time the query is made.

The database also contains planned charge stops P for all charging stations. Each of the planned charge stops $p \in P$ consists of an announced arrival time t^p_{arr} and charging time t^p_{chr}. The charging start time t^p_{start} depends on the arrival time at the charging station (cf. Equation (6)). If we add the charging time, we get the departure time t^p_{dep}:

$$t^p_{start} = t_{start}(t^p_{arr}) \ , \tag{1}$$

$$t^p_{dep} = t^p_{start} + t^p_{chr} \ . \tag{2}$$

A planned charge stop is assigned to the charge point $c \in C_s$, which, regarding its arrival time, would be free soonest:

$$c_p = \underset{c \in C_s}{\operatorname{argmin}}(t^c_{free}(t^p_{arr})) \tag{3}$$

The search iterates over all charge points at the charging station C_s. In practice, this is a very small number so that the search terminates very quickly. The time when a charge point c would be free, from a given arrival time, is returned by function $t^c_{free}(t^p_{arr})$ (cf. Equation (5)) based on information about other planned charge stops in the database.

We denote planned charge stops assigned to the charge point c with an arrival time earlier than the arrival time t_{arr} as:

$$P_c(t_{arr}) = p \in P_s, c_p = c, t_0 < t^p_{arr} < t_{arr} \ . \tag{4}$$

We can now define the time a charge point becomes free for an arrival time t_{arr} as the last departure time of these planned charge stops or, in case there are none, the departure time of the vehicle currently occupying it as

$$t^c_{free}(t_{arr}) = \begin{cases} \max_{p \in P_c(t_{arr})} t^p_{dep} & \text{if} P_c(t_{arr}) \neq \varnothing \\ t^c_{dep} & \text{else} \end{cases} \ . \tag{5}$$

The charging start time is the soonest time a charge point becomes free, but cannot be before the arrival time:

$$t_{start}(t_{arr}) = \max(t_{arr}, \min_{c \in C_s}(t^c_{free}(t_{arr}))) \ . \tag{6}$$

For a queried arrival time t^q_{arr}, we can then simply calculate the charging start time t^q_{start} and thereby the waiting time t^q_{wait}:

$$t^q_{start} = t_{start}(t^q_{arr}) \ , \tag{7}$$

$$t^q_{wait} = t^q_{start} - t^q_{arr} \ . \tag{8}$$

The database also stores statistical data about the utilization of charging stations in the form of average utilization percentage per hour of a day. The information can easily be compiled by regularly querying the current utilization of the charging stations. To account for this statistical utilization in the waiting time estimation, we periodically add short additional charge stops. The period depends on the charging time t_{chr} of the charge stops, the utilization u, and the number of charge points of the charging station n:

$$T = \frac{t_{chr}}{u \cdot n} \ . \tag{9}$$

In our experiments, we set t_{chr} to one minute. This means for a 25 % utilization at a charging station with two charge points, we would add a charge stop every two minutes. An example of the waiting time estimation with additional short charge stops can be seen in Figure 3.

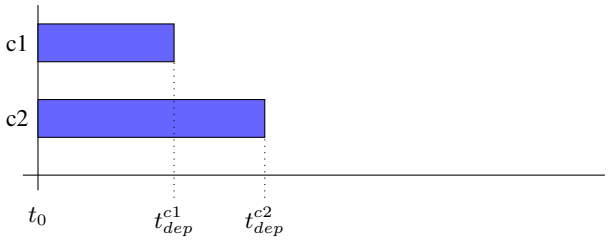
The use of the CSDB is not mandatory, and therefore not all vehicles will announce their charge stops to it. This creates the potential problem, that vehicles that do not use the CSDB and visit a charging station unannounced, may cause significant errors in the waiting time estimation. The statistical data is therefore especially useful, because it is based on past charging station utilization and includes all vehicles regardless of using the CSDB. Additionally, we can update the vehicle's route when the waiting time estimation changes.

The estimated waiting times, which are queried in the beginning of a vehicle's trip, might significantly change by the time the vehicle arrives at the charging stations. Apart from the arrival of unannounced vehicles at charging stations, the change can be due to additional vehicles announcing their planned charge stops to the database. Therefore, to keep the route optimal, we might want to update the route while we are on the trip. Updating the route on the trip requires additional computational resources and communication between vehicles and the CSDB. We have defined three levels of when route updates could take place:

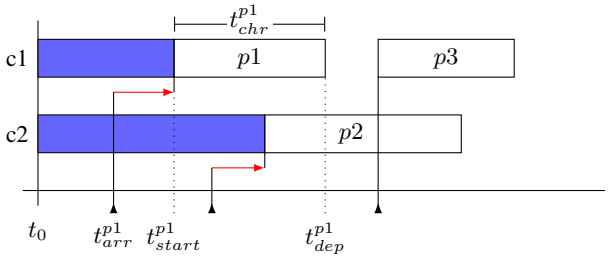
Level 1 The route is only calculated once at the beginning and never updated. The vehicle only communicates with the CSDB once to query waiting time estimates and to announce its planned charge stops.

Level 2 The route is updated every time we arrive at a charging station with the option to skip charging at the charging station. In addition to the initial route calculation, the vehicle communicates with the CSDB at every charge stop to ask for current waiting time estimates. In case of differences to the previous estimates, the route is recalculated. If the planned charge stops have changed as a result, the CSDB has to be informed.

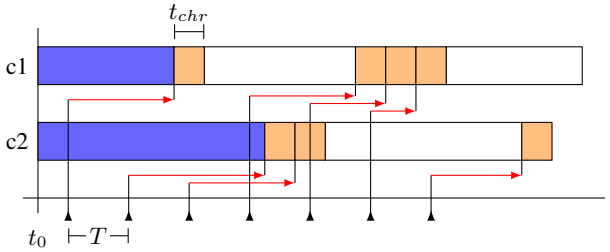
Level 3 The route may be updated at any point in time on the road, i.e., the charging station database informs the vehicle of any changes in the estimated waiting time. The vehicle is in constant communication with the CSDB. In contrast to the first two levels, the CSDB actively checks the waiting time estimates it created for the vehicles and sends updates in case of a change. The vehicles then recalculate their route from the current position on the road. If the planned charge stops have changed as a result, the CSDB has to be informed.



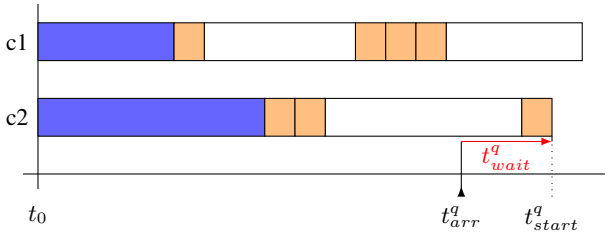
(a) Current utilization of charge points (blue) including departure times of the occupying vehicles



(b) Planned charge stops (white) added to the next free charge point after their arrival (triangle). Waiting times (red arrow) may occur if no charge point is free on arrival



(c) Statistical utilization added as short additional charge stops (orange)



(d) Waiting time of the query t^q_{wait} is the difference between the arrival time t^q_{arr} and the charge start time t^q_{start}

Figure 3. Waiting time estimation example

V. Performance Evaluation

A. Experimental Setup

All experiments were run on a 64 core AMD Ryzen Threadripper 3990X CPU at 2.9 GHz and 256 GB of memory. We implemented our algorithm in C and compiled with GCC 7.5.0 with the highest optimization setting (-O3).

For our experiments, we extracted the road network of Germany from OpenStreetMap excluding very small streets.[1]

[1]Downloaded from download.geofabrik.de on 2020-02-10. All OSM ways with "highway" tag except for path, steps, elevator, corridor, platform, bridleway, footway, cycleway, pedestrian, proposed, construction, raceway, emergency_bay, rest_area, unclassified, residential, living_street, service, tertiary, tertiary_link or track.

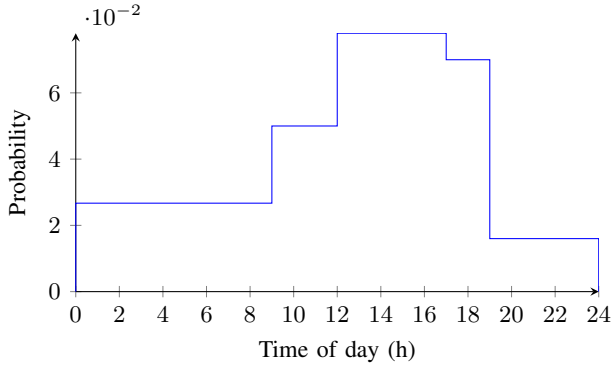


Figure 4. Departure time distribution (based on distribution of trips on a weekday (Mon–Fri) [31])

It has a total of 4,318,497 nodes, of which many only have decorative purposes to model the shape of the road, only 2,356,510 nodes have more than two edges. In the preprocessing step, we contracted 4,317,962 (99.99 %) of the nodes.

We consider the charging stations from the list provided by German Bundesnetzagentur.[2] It contains 9,066 charging stations with a total of 21,780 charge points all over Germany. Because we are only interested in long-distance travel, we used only the fast charging stations of which there are 1,051 with 3,791 charge points, with charging powers ranging from 50–350 kW. We precomputed the shortest-path trees for these charging stations.

For all tests, we simulated one day with a total of 5,000 electric vehicles doing long-distance trips. Each vehicle was assigned an origin/destination pair with a distance of 500 km and a battery capacity ranging from 20–40 kWh, uniformly distributed. This gave the vehicles a range of about 100–300 km, depending on the battery size and their driving speed, ensuring that recharging on the trip was necessary, often multiple times. The departure time was selected from a distribution of trips on a weekday (Mon–Fri) in Germany [31] (see Figure 4). We ran each simulation 10 times and averaged the results.

The simulations are computationally expensive, because as part of the charging station routing we have to evaluate all Pareto optimal paths between the origin, destination and charging stations. By using precomputed shortest-path trees for all charging stations, we only have to explore the graph once for the origin and destination nodes to create shortest-path trees. Querying the Pareto optimal paths is then about two orders of magnitude faster than with plain contraction hierarchies, which would explore the graph for each combination again [9]. However, the query time is still in the order of milliseconds and most queries are between the known locations of the charging stations. To further improve performance, we precomputed all Pareto optimal paths between all combinations of charging stations and stored the costs of the paths in a lookup table. The lookup table with the cost of all Pareto optimal paths between all 1,051 charging stations has a size of 36.3 GB.

[2]https://www.bundesnetzagentur.de/DE/Sachgebiete/ElektrizitaetundGas/Unternehmen_Institutionen/HandelundVertrieb/Ladesaeulenkarte/Ladesaeulenkarte_node.html (visited on 01/04/2020).

The simulation of 5,000 vehicles without using the CSDB took on average 18.6 h or about 13.4 s per vehicle. Vehicles that use the CSDB levels two or three might have to recompute their route many times, increasing computation times. The simulation runs mostly in serial on a single thread, apart from exploring the graph to create shortest-path trees for origin and destination nodes, which is done in parallel. For our tests, we ran 64 simulations simultaneously, with the precomputed shortest-path trees and lookup tables being shared between simulations and held completely in RAM. The total memory usage of all 64 simulations was between 150–200 GB.

B. Energy Consumption Model

An energy model is required to estimate the energy consumption of an electric vehicle. The energy consumption is certainly the most important criterion when it comes to optimize the routing of electric vehicles. For a simplified model, the most important input is the driving speed of the vehicle. The driving speed impacts the energy consumption due to friction and air drag, which are a function of the speed. In addition, other energy consuming components of the vehicle need to be considered, e.g., entertainment system, air conditioning, and the head and tail lights. These components are speed-independent and therefore dominate the energy consumption per km at lower speeds.

We updated our initial energy model [9] to make it more realistic, by fitting the model to the energy consumption of two real electric vehicles. We used data available from Renault [32][3] and Tesla [33]. This resulted in the following energy model:

$$B = 0.05 + \frac{v^2}{90000} + \frac{2}{v} \ . \tag{10}$$

The model is plotted in Figure 5 together with the energy consumption of the two real electric vehicles. As can be seen, our energy model very closely matches their energy consumption on a wide range of speeds. Please note that some factors such as the traffic density or the change in elevation are not incorporated into the model. However, we believe that the accuracy of the model is sufficient to study and evaluate our proposed approach.

C. Battery Charging Model

Traditionally, many authors assumed that the charging speed of a (typical lithium-ion) battery is constant [3], [5]. However, in reality, this speed is very nonlinear after reaching about 80 % of the battery's SOC. It actually decreases considerably at that point [6].

Modern lithium-ion batteries are charged with the CC-CV (constant current - constant voltage) charging protocol [6], [34], [35]. The charging process now follows a two-phase approach. In the first phase, a constant current approach is used for charging the battery. During this time, the charge voltage continuously rises. This process continues until the charge voltage reaches 4.2 V and the SOC is at about 80 %. Now the second phase starts using a constant voltage approach to prevent

[3]Driving range calculator for Renault ZOE with 52 kWh battery. Eco mode off, temperature 20 °C, air conditioner and heater off, 15" wheels

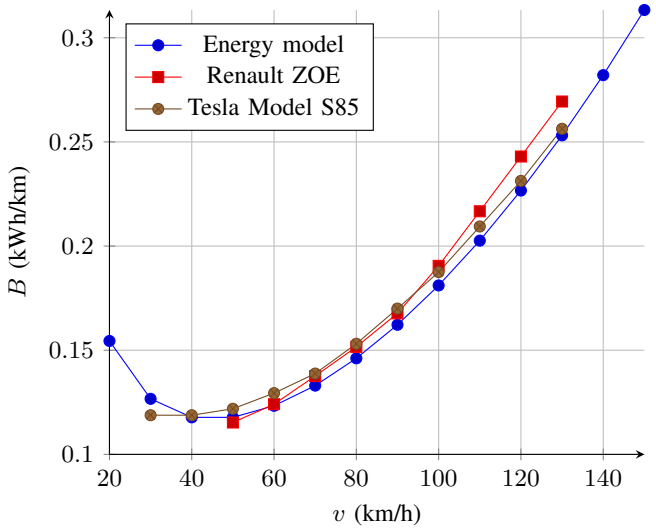


Figure 5. Energy model compared to energy consumption of two real electric vehicles

overcharging. In this phase, the current steadily decreases. The charging process is assumed to be complete when the current falls below a predefined threshold. Alternatively, a CP-CV (constant power - constant voltage) protocol can be used. Here, the charge power is constant in the first phase. Otherwise, it is very similar to the CC-CV approach.

For our purposes, we use the following battery charging model, which supports both the CC-CV and the CP-CV approach. We assume that the voltage increase is linear in the first phase and, for simplicity, the current decrease is also linear in the second phase, which is consistent with the literature [6]. For our model, we use the following variables: The maximum charging power of the charging station is defined as p_{max}. The SOC of the battery is defined as soc in the range $0 \leq soc \leq 1$. In the first phase (constant current/power), the charging voltage increases from $u_{low} = 3.8\,\mathrm{V}$ to $u_{high} = 4.2\,\mathrm{V}$. The phase switch happens exactly at $soc = 0.8$. The maximum current can be calculated as $i_{max} = \frac{p_{max}}{u_{high}}$.

Now, the current $i(soc)$ and voltage $u(soc)$ for the CC-CV charging approach can be calculated based on the SOC of the battery as

$$i(soc) = \begin{cases} i_{max} & \text{for } soc < 0.8 \\ \frac{1-soc}{0.2} \cdot i_{max} & \text{for } soc \geq 0.8 \end{cases} , \tag{11}$$

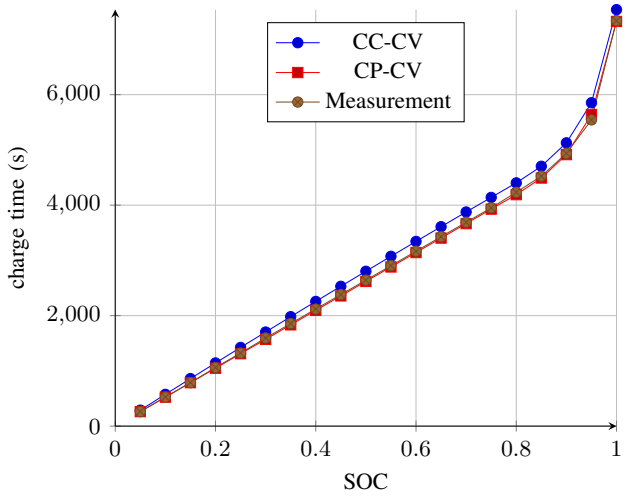
$$u(soc) = \begin{cases} u_{low} + \frac{soc}{0.8}(u_{high} - u_{low}) & \text{for } soc < 0.8 \\ u_{high} & \text{for } soc \geq 0.8 \end{cases} , \tag{12}$$

$$p_{cc\text{-}cv}(soc) = u(soc) \cdot i(soc) \ . \tag{13}$$

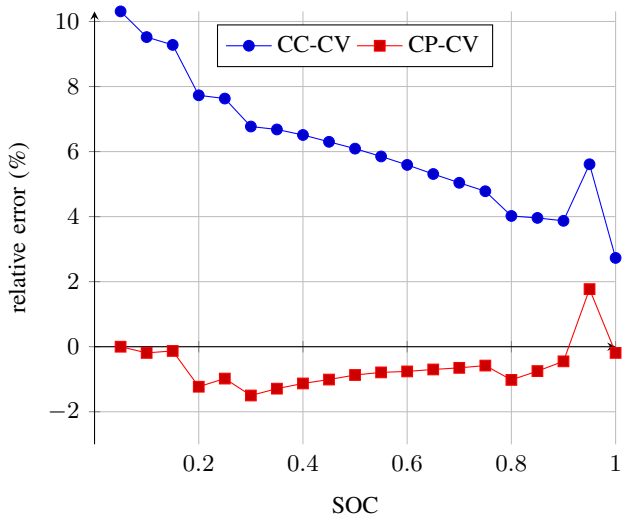
Similarly, the power $p_{cp\text{-}cv}(soc)$ can be calculated as

$$p_{cp\text{-}cv}(soc) = \begin{cases} p_{max} & \text{for } soc < 0.8 \\ u(soc) \cdot i(soc) & \text{for } soc \geq 0.8 \end{cases} . \tag{14}$$

In our algorithm, we estimate the power every second and terminate the charging process when the SOC reaches $soc = 0.99$.



(a) Absolute charge time



(b) Relative error compared to measurement data

Figure 6. Comparison of CC-CV and CP-CV charging protocols with measurement data.

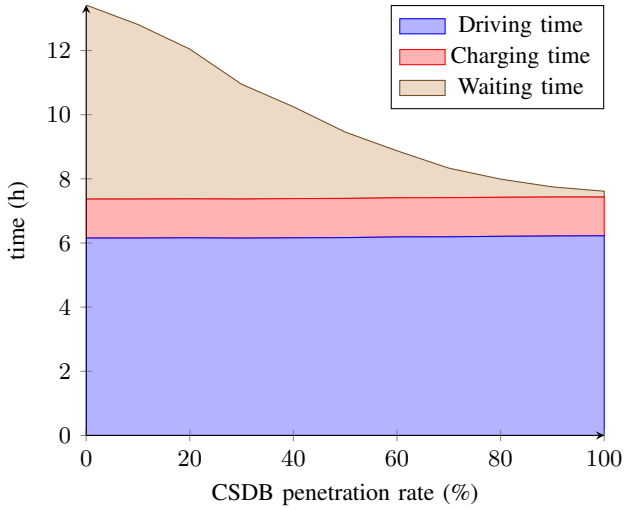


Figure 7. Total travel times of all vehicles for different CSDB penetration rates

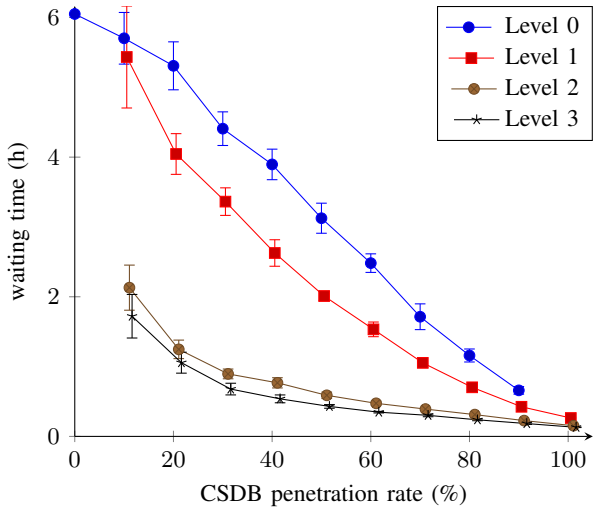


Figure 8. Waiting times of vehicles with different CSDB levels and CSDB penetration rates

In a first validation step, we compared our battery charging model with published measurements of an electric vehicle [36]. The results are shown in Figure 6. Even though the charging protocol is not mentioned for the measurement data, we can see that the CP-CV approach in our model very closely matches the measurement results. Actually, the CP-CV approach has a relative error of ±2 %, whereas the CC-CV protocol has a relative error of more than 10 % at the beginning of the charging process. We conclude that the vehicle was charged using the CP-CV approach.

D. Experiments

In our first experiment, we examined the influence of the percentage of vehicles that use the CSDB (penetration rate) on the total travel time. We tested penetration rates from 0–100 % in 10 % steps. The vehicles using the CSDB were divided equally among the three CSDB levels. As can be seen in Figure 7, the total travel time of all vehicles is reduced significantly by the use of the CSDB. The reduction is mainly due to decreased waiting times at the charging stations, with only small differences in driving time and charging time. Without the CSDB (0 % penetration rate), the average waiting time is 06:03 hours and with all vehicles using it (100 % penetration rate), it is reduced to 11 minutes, which is an improvement of about 97 %.

In Figure 8, we compare the waiting times for the different CSDB levels. For easier comparison, we refer to not using the CSDB, as CSDB level 0. It can be seen, that the CSDB level has a big influence on the average waiting times of the vehicles. At a 10 % penetration rate, levels 0 and 1 are very close with 5.7 h and 5.4 h waiting time, respectively, while levels 2 and 3 have significantly lower waiting times with 2.1 h and 1.7 h, respectively. This is due to the fact, that the initially planned optimal route for vehicles using CSDB level 1, becomes outdated as more and more vehicles announce their planned charge stops. By updating the route at every charge

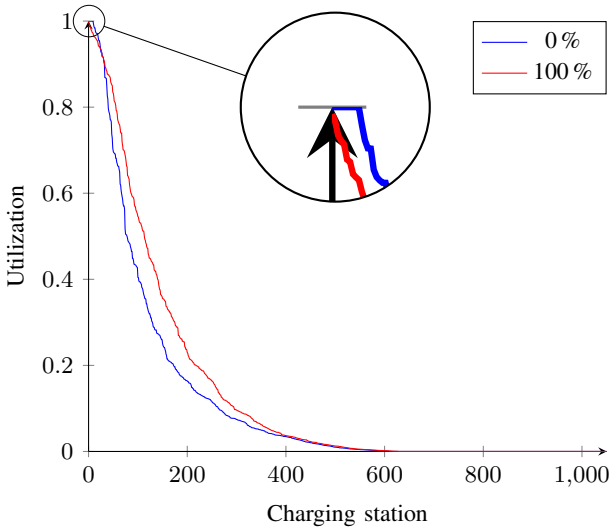


Figure 9. Average utilization of each charging station in peak hours (15–18 h) for CSDB penetration rate of 0 % and 100 %, sorted by utilization

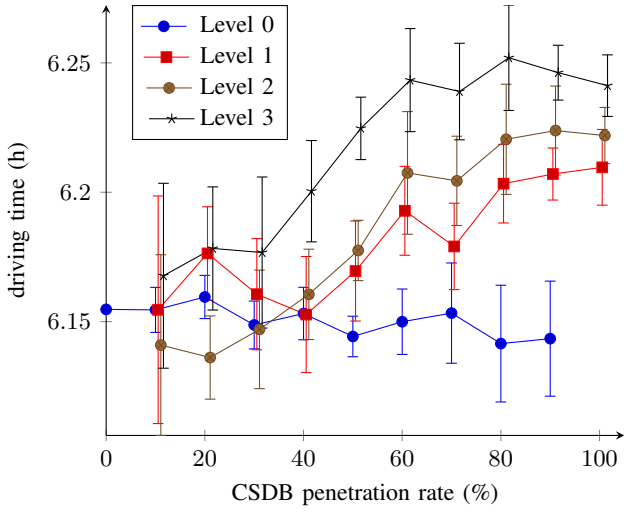


Figure 10. Driving times of vehicles with different CSDB levels and CSDB penetration rates

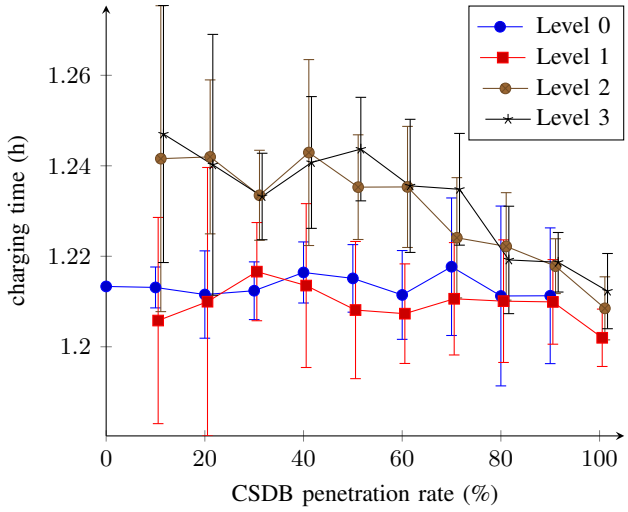


Figure 11. Charging times of vehicles with different CSDB levels and CSDB penetration rates

stop, vehicles using CSDB level 2 improved their waiting times substantially. Vehicles using CSDB level 3 improved slightly over level 2, by updating the route while driving. Obviously, the effect is dependent on the length of the trip and therefore the total travel time. We tested long distances that likely require multiple charge stops. For shorter trips with only one charge stop, we would expect the difference to be much smaller. It can also be observed, that the waiting times of vehicles not using the CSDB also improves significantly with higher penetration rates. The vehicles benefit from the more evenly utilized charging stations, with less hot spots causing long waiting times for all vehicles.

As can be seen in Figure 9, the average utilization during peak hours (15–18 h) at a penetration rate of 0 % is very uneven among the charging stations. The vast majority of charging stations have a low utilization, only a few hot spots of charging stations are utilized 100 % during peak hours and contribute to the long waiting times. At 100 % penetration rate, the load from the hot spots is shifted to other charging stations, making the utilization of charging stations more even and reducing waiting times significantly. Even then, the charging stations are very unevenly utilized. Part of the reason are big differences in charging speed and many charging stations not being close to highways, making them undesirable for long-distance travel. As we focus only on long-distance travel in our simulation, the utilization by vehicles doing shorter trips is not considered.

The driving times and charging times are also slightly influenced by using the CSDB. As can be seen in Figure 10, the average driving time correlates with a higher penetration rate and a higher CSDB level. As more vehicles use the CSDB, the vehicles have more information about planned charge stops of other vehicles in the future and are therefore more likely to drive detours to alternative charging stations with lower waiting times. This is especially true for vehicles using higher CSDB levels, as they can update their route, which may lead to additional detours based on planned charge stops that have been announced in the meantime. This is not a problem, as the additional driving time is only small and is more than compensated by the saved waiting time. The average charging times can be seen in Figure 11. Vehicles using CSDB levels 2 and 3 have a slightly higher charging time at low penetration rates. This is caused by the vehicles changing their selected charging stations to less optimal ones, in case there are other unexpected vehicles at the selected charging station. As more vehicles use the CSDB, encountering unexpected vehicles becomes less likely.

In our second experiment, we compared using statistical data about the charging station utilization with only using the current utilization and planned charge stops. We generated the statistical data from the charging station utilization of simulation runs of the first experiment. In Figure 12, we compare the average waiting times with and without using statistics for vehicles using CSDB Level 3. It can be seen that using statistical data approximately halves the waiting times.

Interestingly, it not only improves the waiting times for low

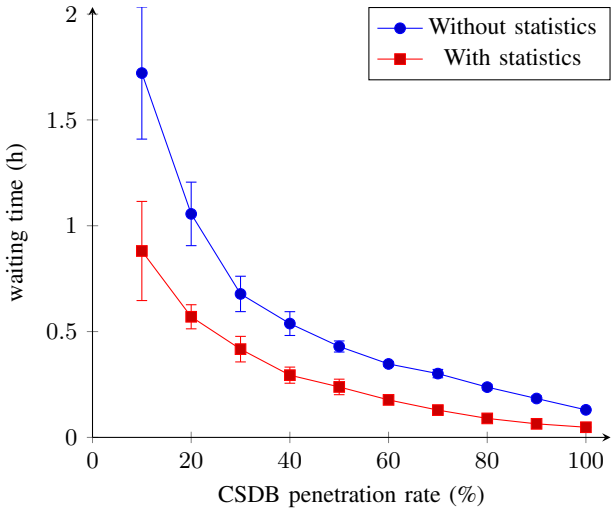
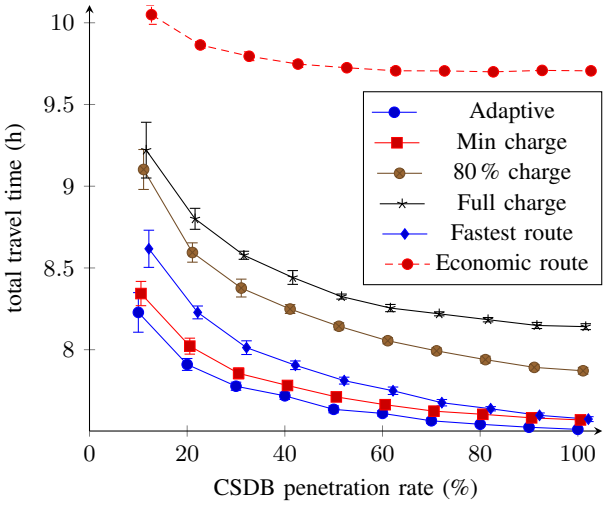
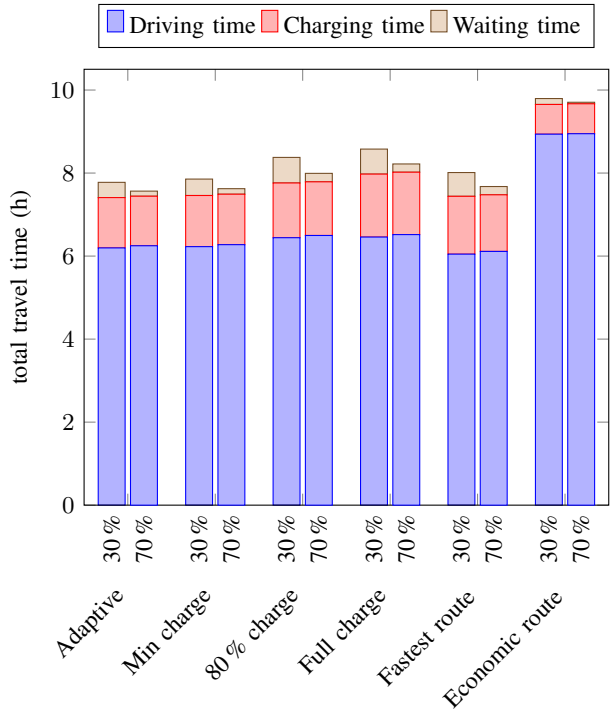


Figure 12. Waiting times compared with and without using statistics on CSDB Level 3



(a) Total travel time for different CSDB penetration rates



(b) Total travel time composition at 30 % and 70 % penetration rate

Figure 13. Comparison of adaptive charging and routing strategy to other strategies (CSDB level 3 with statistics)

penetration rates, where many vehicles are not accounted for by the CSDB, but also for high penetration rates including 100 %, where all vehicles announce their planned charge stops. When a vehicle initially plans its trip at departure time, the CSDB will only know about planned charge stops of vehicles that have already departed. Even though the vehicle can update its route while driving, by the time it becomes known that it is on a suboptimal path, it might already be too late to change it. It is therefore beneficial to account for vehicles departing in the future by using statistical data.

In our third experiment, we compared our adaptive charging and routing strategy with alternative strategies often found in literature. For each strategy, we ran a set of simulations with vehicles using CSDB level 3 and statistics at penetration rates 10–100 % in 10 % steps. In Figure 13, the total travel time of the strategies over all penetration rates and the composition of the total travel time for the penetration rates 30 % and 70 % are shown.

We can see that using the adaptive strategy results in the least total travel time in all cases. It is very closely followed by always doing a minimum charge, just to get to the next charging station (min charge). The adaptive charging strategy only has an advantage over min charge in cases where the first charge is at a fast charging station and then a second charge is at a slower charging station. As there are plenty of fast charging stations available, this is a rare case. Always doing an 80 % charge or always doing a full charge, causes significantly more charging time, because more energy than necessary is being charged. This makes choosing a fast charging station more important to the algorithm, which results in additional driving time due to detours and additional waiting time due to higher utilizations at fast charging stations.

The adaptive routing strategy selects the route out of the set of Pareto optimal routes that minimizes driving, charging and waiting time. We compared this with always choosing the fastest route and always choosing the most economic route. Always choosing the fastest route reduces the driving time, but leads to more charging time due to more energy consumption, which in turn leads to more waiting time due to higher charging station utilization. The opposite can be observed when always choosing the most economic route. The charging time is reduced significantly due to less energy consumption, which also leads to a very short waiting time due to lower charging station utilization, but the driving time is significantly longer, making it the strategy with the highest total travel time.

In all cases, we can see that a higher penetration rate of the CSDB reduces the waiting time and the total travel time.

VI. Conclusion

We presented an approach to minimize waiting times at charging stations for long-distance trips with electric vehicles by announcing planned charge stops to a central charging station database (CSDB). We integrated the waiting time estimates of the CSDB into our adaptive charging station routing approach to minimize the total travel time of electric vehicles. In our evaluation, we considered the map of Germany and simulated one day with a large number of vehicles doing long-distance trips. The existing heterogeneous charging infrastructure with its differences in charge power and number of charge points was used together with a realistic non-linear charge model. We showed that the utilization of charging stations is very uneven and can cause long waiting times, but that by using the CSDB, average waiting times can be reduced by up to 97 %. As the waiting time estimates can become outdated after a while, the route of long-distance trips should be updated at least at every charge stop. Updating the route while driving further improves the waiting time. By using statistical data about the utilization of the charging stations derived from historical data, we could additionally reduce the waiting time by about half. Furthermore, we compared using the CSDB with our adaptive charging and routing strategies to other strategies often found in literature. We could clearly show, that while using the CSDB is beneficial to reducing waiting times in all cases, the combination with our adaptive strategies provides the best total travel times.

In future work, we want to better take human behavior and individual preferences into account. Human drivers might be frustrated by frequent route changes or need to rest some time after a long drive. We mentioned in Section IV that the CSDB is different than a reservation system and that we assume that it is more flexible when dealing with early or late arrivals. Further studies are needed to verify that assumption by making a thorough comparison between the approaches. We also want to look into using the CSDB and adaptive strategies for short-distance trips where recharging is not necessary to reach the destination, but the driver simply wants to recharge the vehicle and could either stop at a charging station on the way or charge near his destination, which might entail additional walking time. In this context, we want to compare having many slow charging stations to having few fast charging stations. Additionally, we want to improve the energy consumption and battery charging models. Our current universal model does not fit all existing vehicles equally well. There are many different electric vehicle models on the market, which can have very different energy consumption profiles and charge curves, especially for fast charging. We therefore want to incorporate individual vehicle models with model specific charge curves and energy consumption profiles.

References

[1] IEA, "Global EV Outlook 2020," IEA, Paris, Tech. Rep., 2020.

[2] B. Berman. (2019). Rapid EV adoption brings queues at some EV chargers on Thanksgiving weekend, [Online]. Available: https://electrek.co/2019/12/01/rapid-ev-adoption-brings-queues-at-some-ev-chargers-on-thanksgiving-weekend/ (visited on 08/27/2020).

[3] M. Alizadeh, H.-T. Wai, A. Scaglione, A. Goldsmith, Y. Fan, and T. Javidi, "Optimized Path Planning for Electric Vehicle Routing and Charging," in *52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton 2014)*, Monticello, IL: IEEE, Sep. 2014, pp. 25–32.

[4] 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, Mar. 2019.

[5] M. Baum, J. Sauer, D. Wagner, and T. Zündorf, "Consumption Profiles in Route Planning for Electric Vehicles: Theory and Applications," in *16th International Symposium on Experimental Algorithms (SEA 2017)*, London, United Kingdom: Schloss Dagstuhl, Jun. 2017, 19:1–19:18.

[6] K. S. Ng, C.-S. Moo, Y.-P. Chen, and Y.-C. Hsieh, "Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries," *Applied Energy*, vol. 86, no. 9, pp. 1506–1511, Sep. 2009.

[7] M. M. De Weerdt, S. Stein, E. H. Gerding, V. Robu, and N. R. Jennings, "Intention-Aware Routing of Electric Vehicles," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 17, no. 5, pp. 1472–1482, May 2016.

[8] M. T. Goodrich and P. Pszona, "Two-Phase Bicriterion Search for Finding Fast and Efficient Electric Vehicle Routes," in *22nd ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL 2014)*, Dallas-Fort Worth, TX: ACM, Nov. 2014, pp. 193–202.

[9] S. Schoenberg and F. Dressler, "Planning Ahead for EV: Total Travel Time Optimization for Electric Vehicles," in *22nd IEEE International Conference on Intelligent Transportation Systems (ITSC 2019)*, Auckland, New Zealand: IEEE, Oct. 2019.

[10] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[11] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, Apr. 1968.

[12] R. Geisberger, P. Sanders, D. Schultes, and D. Delling, "Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks," in *7th International Workshop on Experimental Algorithms (WEA 2008)*, Provincetown, MA: Springer, May 2008, pp. 319–333.

[13] M. Baum, J. Dibbelt, D. Wagner, and T. Zündorf, "Modeling and Engineering Constrained Shortest Path Algorithms for Battery Electric Vehicles," *Transportation Science*, Oct. 2020.

[14] A. Artmeier, J. Haselmayr, M. Leucker, and M. Sachenbacher, "The Shortest Path Problem Revisited: Optimal Routing for Electric Vehicles," in *Advances in Artificial Intelligence (KI 2010)*, Karlsruhe, Germany: Springer, Sep. 2010, pp. 309–316.

[15] S. Storandt, "Quick and Energy-Efficient Routes - Computing Constrained Shortest Paths for Electric Vehicles," in *5th ACM SIGSPATIAL International Workshop on Computational Transportation Science (IWCTS 2012)*, Redondo Beach, CA: ACM, Nov. 2012, pp. 20–25.

[16] ——, "Route Planning for Bicycles - Exact Constrained Shortest Paths Made Practical Via Contraction Hierarchy," in *22nd International Conference on Automated Planning and Scheduling (ICAPS 2012)*, Atibaia, Brazil: AAAI Press, Jun. 2012, pp. 234–242.

[17] J. Dibbelt, T. Pajor, and D. Wagner, "User-Constrained Multi-Modal Route Planning," in *Meeting on Algorithm Engineering & Experiments (ALENEX 2012)*, Kyoto, Japan: Society for Industrial and Applied Mathematics, Jan. 2012, pp. 118–129.

[18] F. Hartmann and S. Funke, "Energy-Efficient Routing: Taking Speed into Account," in *Advances in Artificial Intelligence (KI 2014)*, Stuttgart, Germany: Springer, Sep. 2014, pp. 86–97.

[19] 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 (TITS)*, pp. 1–13, 2019.

[20] Z. Sun and X. Zhou, "To save money or to save time: Intelligent routing design for plug-in hybrid electric vehicle," *Transportation Research Part D: Transport and Environment*, vol. 43, pp. 238–250, Mar. 2016.

[21] H.-J. Kim, J. Lee, G.-L. Park, M.-J. Kang, and M. Kang, "An Efficient Scheduling Scheme on Charging Stations for Smart Transportation," in *1st International Conference on Security-Enriched Urban Computing and Smart Grid (SUComS 2010)*, Daejeon, South Korea, Sep. 2010, pp. 274–278.

[22] H. Qin and W. Zhang, "Charging Scheduling with Minimal Waiting in A Network of Electric Vehicles and Charging Stations," in *8th ACM*

International Workshop on Vehicular Internetworking (VANET 2011), Las Vegas, NV: ACM, Sep. 2011.

[23] L. Bedogni, L. Bononi, M. Di Felice, A. D'Elia, and T. S. Cinotti, "A Route Planner Service with Recharging Reservation: Electric Itinerary with a Click," *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 3, pp. 75–84, 2016.

[24] Y. Cao, S. Liu, Z. He, X. Xie, R. Wang, and S. Yu, "Electric Vehicle Charging Reservation Under Preemptive Service," in *1st International Conference on Industrial Artificial Intelligence (IAI 2019)*, Shenyang, China, Jul. 2019.

[25] L. Hou, C. Wang, and J. Yan, "Bidding for Preferred Timing: An Auction Design for Electric Vehicle Charging Station Scheduling," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 21, no. 8, pp. 3332–3343, Aug. 2020.

[26] K.-B. Lee, M. A. Ahmed, D.-K. Kang, and Y.-C. Kim, "Deep Reinforcement Learning Based Optimal Route and Charging Station Selection," *Energies*, vol. 13, no. 23, p. 6255, Nov. 2020.

[27] T. Qian, C. Shao, X. Wang, and M. Shahidehpour, "Deep Reinforcement Learning for EV Charging Navigation by Coordinating Smart Grid and Intelligent Transportation System," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1714–1723, Mar. 2020.

[28] C. Zhang, Y. Liu, F. Wu, B. Tang, and W. Fan, "Effective Charging Planning Based on Deep Reinforcement Learning for Electric Vehicles," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 22, no. 1, pp. 542–554, Jan. 2021.

[29] S.-N. Yang, W.-S. Cheng, Y.-C. Hsu, C.-H. Gan, and Y.-B. Lin, "Charge scheduling of electric vehicles in highways," *Mathematical and Computer Modelling*, vol. 57, no. 11, pp. 2873–2882, Jun. 2013.

[30] F. Dressler, G. S. Pannu, F. Hagenauer, M. Gerla, T. Higuchi, and O. Altintas, "Virtual Edge Computing Using Vehicular Micro Clouds," in *IEEE International Conference on Computing, Networking and Communications (ICNC 2019)*, Honolulu, HI: IEEE, Feb. 2019.

[31] G. Pasaoglu, D. Fiorello, A. Martino, G. Scarcella, A. Alemanno, A. Zubaryeva, and C. Thiel, "Driving and parking patterns of European car drivers - A mobility survey," Institute for Energy and Transport (Joint Research Centre), European Commission, Scientific and Technical Research Reports, Nov. 2012.

[32] Renault. (2020). Reichweite, Batterie und Aufladen Renault ZOE, [Online]. Available: https://www.renault.de/elektrofahrzeuge/zoe/batterie-und-laden.html (visited on 07/01/2020).

[33] E. Musk and J. B. Straubel. (2012). Model S Efficiency and Range, [Online]. Available: https://www.tesla.com/blog/model-s-efficiency-and-range (visited on 07/01/2020).

[34] M. Etezadi-Amoli, K. Choma, and J. Stefani, "Rapid-Charge Electric-Vehicle Stations," *IEEE Transactions on Power Delivery*, vol. 25, no. 3, pp. 1883–1887, Jul. 2010.

[35] S. Bashash, S. J. Moura, J. C. Forman, and H. K. Fathy, "Plug-in hybrid electric vehicle charge pattern optimization for energy cost and battery longevity," *Journal of Power Sources*, vol. 196, no. 1, pp. 541–549, Jan. 2011.

[36] T. Zündorf, "Electric Vehicle Routing with Realistic Recharging Models," Master's Thesis, Department of Informatics, Karlsruhe, Germany, Nov. 2014.

Sven Schoenberg received his B.Sc. degree from Dortmund University of Applied Sciences and Arts, Germany together with a B.Sc. (Hons) degree from Staffordshire University, U.K. in 2011 (double degree), and a M.Sc. degree from Paderborn University, Germany in 2014. He is currently working for the Software Innovation Lab at Paderborn University and is an external Ph.D. Candidate in the Data Communications and Networking Group of Falko Dressler at TU Berlin. His research focuses on algorithms for electric vehicle route planning.

Falko Dressler received his M.Sc. and Ph.D. degrees from the Dept. of Computer Science, University of Erlangen in 1998 and 2003, respectively. He is a full professor and Chair for Data Communications and Networking at the School of Electrical Engineering and Computer Science, TU Berlin. Dr. Dressler has been associate editor-in-chief for IEEE Trans. on Mobile Computing and Elsevier Computer Communications as well as an editor for journals such as IEEE/ACM Trans. on Networking, IEEE Trans. on Network Science and Engineering, Elsevier Ad Hoc Networks, and Elsevier Nano Communication Networks. He has been chairing conferences such as IEEE INFOCOM, ACM MobiSys, ACM MobiHoc, IEEE VNC, IEEE GLOBECOM. He authored the textbooks Self-Organization in Sensor and Actor Networks published by Wiley & Sons and Vehicular Networking published by Cambridge University Press. He has been an IEEE Distinguished Lecturer as well as an ACM Distinguished Speaker. Dr. Dressler is an IEEE Fellow as well as an ACM Distinguished Member. He is a member of the German National Academy of Science and Engineering (acatech). He has been serving on the IEEE COMSOC Conference Council and the ACM SIGMOBILE Executive Committee. His research objectives include adaptive wireless networking (radio, visible light, molecular communications) and embedded system design (from microcontroller to Linux kernel) with applications in ad hoc and sensor networks, the Internet of Things, and cooperative autonomous driving systems.