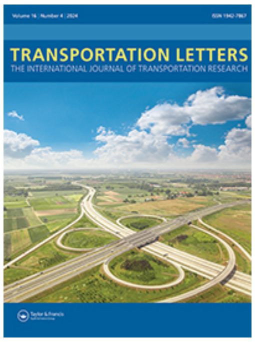


Transportation Letters

The International Journal of Transportation Research

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/ytrl20

# Route and charging planning for electric vehicles: a multi-objective approach

Marek Cuchý, Michal Jakob & Jan Mrkos

TRANSPORTATION LETTERS
https://doi.org/10.1080/19427867.2024.2315359

Taylor & Francis
Taylor & Francis Group



# Route and charging planning for electric vehicles: a multi-objective approach

Marek Cuchý, Michal Jakob and Jan Mrkos

Artificial Intelligence Center, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic

**ABSTRACT**

Electric vehicle (EV) travel planning is a complex task that involves optimizing both the routes and the charging sessions for EVs. Existing algorithms rely on single-objective optimization, which limits their ability to consider EV users' multiple, often conflicting objectives. In this paper, we introduce a new, genuinely multi-objective approach to EV travel planning, which can find Pareto sets containing multiple EV travel plans optimized simultaneously for multiple objectives. We focus on the bi-objective optimization for travel time and cost. To our knowledge, our algorithm is the first to perform such a genuine multi-objective optimization on realistically large country-scale problem instances involving 12,000 charging stations. We implemented our approach into a fully operational prototype application and extensively evaluated it on real-world data. Our results show that our approach can achieve practically usable planning times with only a minor loss of solution quality despite the very high computational complexity of the problem.

**ARTICLE HISTORY**
Received 21 August 2023
Accepted 1 January 2024

**KEYWORDS**
Route planning; electric vehicles; charging cost; charging stations; speed-up techniques

## Introduction

Today's EV drivers are not overly concerned with their vehicle's range. Most EVs can travel more than 200 km on a single charge, a range sufficient for daily commutes. However, a recurring concern of EV drivers today is planning long-distance trips with charging at unknown locations.[1]

A user planning a trip from Passau to Hamburg in Germany (around 800 km journey) can check online whether the charging stations (CSs) along their route are working and their price, and use satellite navigation to drive between the CSs. Experienced users will use a dedicated application[2] that will find the fastest route with charging stops. However, the user cannot easily determine how much the trip will cost or whether there are cheaper options. The same trip may cost half if the driver selects a different charging station with a slight detour. Therefore, a trip planned by a route planner that optimizes only travel time might be rather costly.

Weighing trade-offs between price and speed is not a new problem in general transportation (e.g. bus vs. airplane), but it is not something most drivers had to consider in the past. However, this shift in planning behavior could be achieved seamlessly if EV drivers had a tool that plans their route (including charging stops) and presents them with different options for travel time and cost. The user could then select an option best fitting their needs. In this paper, we solve this *multi-objective EV travel planning*[3] problem with practically applicable solution times on country-sized road graphs. To demonstrate the practical usefulness of our solution, we have set up a prototype application,[4] see Figure 1 for an example solution provided by the application.

Multi-objective EV travel planning is a complex problem (NP-hard but not even in NP) for two main reasons that require both domain-independent and domain-specific techniques to overcome:

(1) *Multi-objective optimization*: The problem involves multiple, inherently conflicting objectives (travel time and cost), which inflates the dimensionality of the search space and extends the solution concept from a single solution (route with charging stops) into a Pareto-set of solutions, each with different trade-off of duration and cost).
(2) *Integration of charging planning with route planning*: The EV travel planning problem is actually composed of two subproblems – planning the route in the road network and choosing where and how long to charge. These two problems are closely interconnected, and therefore, we need to solve them holistically to obtain the best solutions.

The multi-objective EV route planning problem we address in this paper is further complicated but, at the same time, more applicable in practice by our use of realistic battery charging (non-linear function), large road networks (country-sized), and different prices and speeds of charging at different CSs.

Despite the significant progress in *EV travel planning* in recent years, the existing algorithms cannot yet fully support the above-described use case. The main limitation of existing approaches is that they mostly rely on *single-objective* optimization (e.g. Baum, Dibbelt, Gemsa, et al. 2019) and are therefore technically limited to always considering only a single objective when finding optimal EV travel plan. Well-established approaches to multi-objective optimization, such as meta-heuristics, can find the Pareto-set only on very small city-sized road networks. Consequently, these approaches are *not* suitable in practice (e.g. Ben Abbes, Rejeb, and Baati 2022). Very recent work of Schoenberg and Dressler (2023) achieved good planning times while considering multiple simpler objectives (not including cost) on country-scale road networks. Although their approach is similar to ours, it prohibits planning with a realistic number of charging stations.

In this paper, we, therefore, introduce a new, more powerful *multi-objective* approach to EV travel planning for a single EV between an origin and an EV user's destination with multiple charging stops. Our main contributions are:

**CONTACT** Marek Cuchý marek.cuchy@fel.cvut.cz Artificial Intelligence Center, Faculty of Electrical Engineering, Czech Technical University in Prague, Charles Square 13, Prague, Czech Republic

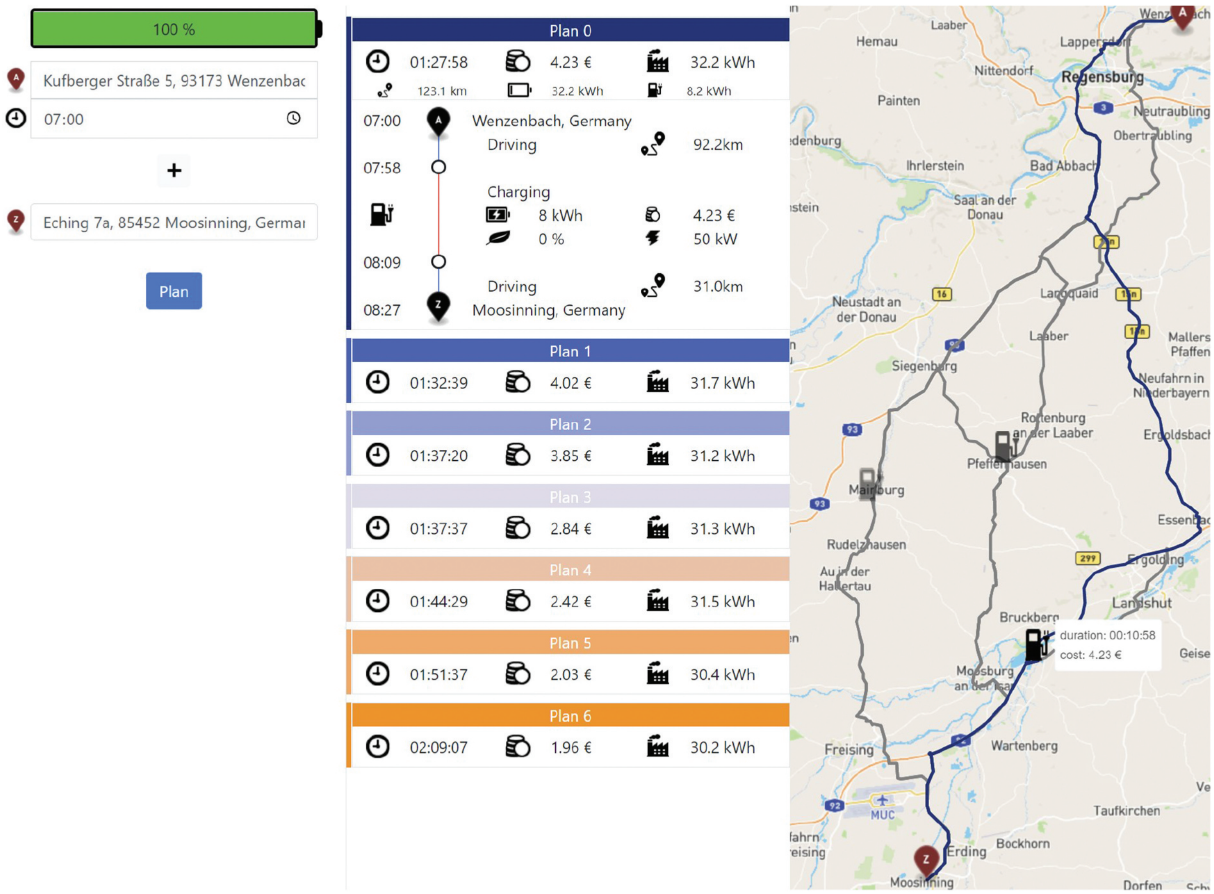


**Figure 1.** Screenshot of the prototype application implementing the EV travel planning algorithms described in the paper. The application backend is the same implementation used for the evaluation, although running on a slower HW. The planning request can be specified in the left panel, the summary of found plan variants and the detail of a selected plan can be seen in the center, and the right panel displays the plans on the map.

(1) We formalized a novel, practically relevant variant of a Multi-Objective EV Travel Planning Problem.
(2) We designed two new, practically applicable, multi-objective algorithms capable of solving the EV travel planning problem on large road networks. The first, *optimal* algorithm can find a complete Pareto-optimal set of EV travel plans, while the second, *approximate* algorithm can find near-optimal Pareto-sets in mere seconds, allowing to be tuned for different trade-offs between speed and accuracy.
(3) We performed an extensive empirical evaluation of the proposed problem parameters and algorithms on real-world country-scale data with more than 12 000 charging stations involving 780 000 calculated problem instances requiring more than 2 million CPU hours.

To the best of our knowledge, ours is *the first EV travel planning algorithm* capable of performing such a genuine multi-objective optimization on realistically large problem instances (country-sized road networks with more than 12,000 charging stations).

Importantly, our proposed approach is very versatile and can be adapted to other optimization objectives and more complex scenarios, such as time-dependent travel times and charging prices, and therefore presents a generic approach to solving a wide range of multi-objective EV travel planning problems. We believe our contribution will provide a solid basis for the future exploration of multi-objective approaches to EV travel planning.

The paper is structured as follows. In Related work section, we discuss related problems and highlight how our solutions differ from other approaches to EV travel planning described in recent literature. Multi-objective EV travel planning problem section specifies the multi-objective EV route planning problem we are solving, including the EV model we use. Next, we describe the Multi-objective EV travel planning algorithm in detail. Finally, Evaluation setup section describes the extensive experiments used to evaluate our problem and algorithms, and we also provide discussion in Evaluation results section. Specifically, we highlight how the different proposed speed-ups and problem parameters influence the performance of the algorithms.

## Related work

Our problem formulation can be viewed as a *multi-objective* route planning problem on large road networks *constrained* by the short-range EV battery and the resulting need to stop at *charging* stations. Below, we give an overview of the relevant approaches, starting from the general algorithms for general graphs. These form the basis of most route planning solutions. We also describe route planning[5] algorithms historically developed for combustion engine vehicles and how they are limited for use in EV planning. Then, we focus on many versions of single-objective EV travel planning problems and the difficulties they pose. Many of which our proposed approach tackles. Finally, we describe other multi-objective approaches to planning within the context of EVs and how our work extends the state of the art.

### Shortest path on generic graphs

Most vehicle route planning algorithms have their origins in algorithms for solving the standard single-objective shortest path problem (problem in P), which can be traced as far back as to Dijkstra's algorithm (Dijkstra 1959).

Constrained shortest path problems (Aneja and Nair 1978) extend the classical shortest path problem to a situation when the path needs to fulfill additional constraints. The ability to handle constraints is essential for EV route planning because of the need to keep the EV state of charge within valid bounds. In contrast with the standard, unconstrained single-objective shortest path problem, constrained shortest path problems are generally NP-hard (Garey and Johnson 1979).

The multi-objective version of the unconstrained shortest path problem has been first addressed by the label-setting algorithm (Martins 1984). The multi-objective label-setting algorithms need to maintain a whole set of Pareto-optimal labels in their open queue – instead of a single label per node used by single-objective algorithms. The number of paths in the Pareto-set can be exponential in the size of the graph, making the problem inherently very hard (NP-hard but not even in NP) (Müller-Hannemann and Weihe 2006). Classical informed-search A* algorithm (Hart, Nilsson, and Raphael 1968) has been extended to the multi-objective, unconstrained setting in the MOA* algorithm (Stewart and White 1991) and later improved by NAMOA* algorithm (Mandow et al. 2005). All these multi-objective algorithms can be straightforwardly modified to solve the constrained version.

Based on the above, the EV travel planning problem is difficult to solve. However, road graphs used in routing have special properties that make the shortest path search more feasible than in generic graphs.

### Route planning on road transport networks

Moving from generic graphs to graphs representing road networks, researchers focused on developing a wide range of search speed-ups exploiting the specific hierarchical, quasi-planar structure of road transport networks (Eppstein and Goodrich 2008). Decades of research resulted in speed-ups as high as $10^6$ compared to the baseline Dijkstra's algorithm, enabling sub-millisecond route planning times on continental-sized road networks (Bast et al. 2016).

Unfortunately, except for a few exceptions (e.g. Hrnčíř et al. 2016; Zhu 2022), which study multi-objective bicycle routing, the vast majority of research on road network route planning deals with the single-objective, unconstrained formulation of the route planning problem and, as such, it is not directly applicable for EV travel planning.

A potentially applicable method is presented by Delling and Wagner (2009). The work proposes a multi-objective adaptation of SHARC algorithm (Bauer and Delling 2009) that combines highway hierarchies (Sanders and Schultes 2006) and arc-flags (Möhring et al. 2007) techniques. However, the arc-flag technique is unsuitable for planning with charging stops. This is because the arc-flag technique divides the graph into many partitions with a small number of boundary edges. It assigns a set of flags to each boundary edge that says if the edge lies on the shortest path to other partitions (to at least one node in the given partition). Such information is hard to utilize since optimal EV plans detour from the shortest path to recharge the battery at charging stations.

### Planning for electric vehicles

EV-specific planning has become an active area of research in the past decade. Researchers have studied many variants of EV planning problems, differing in, e.g. whether energy consumption is treated as an optimization objective or considered, in conjunction with the state of charge (SoC) of the EV battery, only as a constraint.

One of the first approaches focused on finding the most energy-efficient routes without considering charging stops. Although this is the simplest variant of the EV planning problem, it still prohibits the use of the label-setting Dijkstra's algorithm because of the possible presence of negative edges due to energy recuperation. To circumvent this problem, Artmeier et al. (2010) proposed a solution utilizing the Bellman-Ford algorithm (Bellman 1958) and the label-correcting version of Dijkstra's algorithm. Sachenbacher et al. (2011) propose a different approach to address the presence of negative edges; their approach first removes the negative edges by the *potential shifting* technique (Johnson 1977) and then runs an A* search on the newly created graph. Schönfelder et al. (2014) extend the problem of finding the most energy-efficient route by searching not only for a single solution for a given initial SoC but rather for the *consumption profile* piecewise linear function that computes the optimal consumption and route for any possible initial SoC.

However, the mid-trip charging stops are essential for planning trips exceeding the range of the EV battery and/or optimizing the EV charging over multiple days. The first algorithm capable of planning charging in EV trips was proposed by Storandt and Funke (2012). The limitation of this work is that the EV is always charged to the full battery capacity. This assumption significantly simplifies the problem.

The problem with planning EV charging stops is that there is virtually an infinite number of target SoCs to consider for each visited charging station. The A*-based algorithm finding the most energy-efficient EV travel plan proposed by Baum et al. (2019b) addresses this issue with charging by exploiting *consumption profiles* between charging stops, which allows to significantly reduce the number of generated planning states while still maintaining optimality. The core idea of the algorithm is that the generation of labels representing various charging options at a charging station is postponed until the next charging station. This allows the algorithm to leverage the additional information about the consumption between the two charging stations to significantly reduce the originally infinite number of potentially generated labels.

The author also extended this approach in (Baum, Dibbelt, Gemsa, et al. 2019) and designed an algorithm that solves the problem with SoC only as a constraint and travel time as the objective and finding the shortest feasible EV travel plan (also studied by Storandt 2012). Baum et al. (2019a) also employs realistic charging models that were first extensively studied by Zündorf (2014). Charging models are important because the time required to charge an EV usually differs significantly based on the initial SoC. For example, it is usually much slower to charge the EV battery from 80% to 100% than it is from 20% to 40%. In these works, the authors model this behavior with a charging function that could be, for example, piecewise linear. However, since the above approaches consider only a single objective, the trade-offs between the optimization objectives are ignored.

The real-world energy consumption of an EV is highly dependent on many difficult-to-estimate variables, such as the driving style of the driver and the condition of the vehicle. Therefore, Rajan et al. (2021) and Ünal et al. (2022) propose approaches that take into account this uncertainty.

### Multi-objective planning for electric vehicles

Common approaches for solving multi-objective problems, such as genetic algorithms (Ben Abbes, Rejeb, and Baati 2022) or particle swarm optimization (Siddiqi, Shiraishi, and Sait 2011), were applied to EV travel planning. Although the authors consider the cost of charging in these works, the methods were evaluated only on very small road networks with only hundreds of nodes. Realistic road graphs required in EV route planning have millions of nodes. As such, these techniques do not currently scale to realistic problem instances.

Genetic algorithms and other meta-heuristics are also used for more complex multi-objective problems where the route and charging in EV planning are solved only as sub-problems, such as distribution of charging stations (Tran et al. 2021) or design of the electric public transit network (Liu et al. 2020). However, individual EV planning in these problems is often oversimplified, so the results cannot be used in practice by EV drivers. For example, they commonly simplify the routing part, where they consider only one pre-calculated route between charging stations and/or other points of interest (e.g. bus stops). Furthermore, the graphs used in these works contain orders of magnitude less nodes than needed for practical applications.

Schoenberg and Dressler (2023) proposed an algorithm based on multi-objective A* and a set of speed-ups that finds a Pareto-optimal set of plans while minimizing time and energy consumption. Although the proposed algorithm can find the Pareto-set in milliseconds, the solved problem is much simpler. In our proposed problem, we need to consider three different attributes to be non-dominated – two objectives (time and cost) and one constraint (energy consumption). The problem proposed by Schoenberg and Dressler (2023) requires only two attributes to be non-dominated – two objectives (time and energy consumption), where the latter is also a constraint. The dimension of the non-dominated attributes has the greatest impact on the complexity of a multi-objective problem. Moreover, one of the proposed speed-ups precalculate the routes between all pairs of charging stations, which is possible if the number of charging stations is small. The authors considered approx. 1000 charging stations in their evaluation, and the data has a size of 36GB. Since the size requirements are quadratic, we can assume that the 12 thousand charging stations, we use in our experiments, would require approx. 244× more space, making this speed-up unusable in our problem.

### Summary

Despite the proliferation of work addressing different aspects of EV travel planning, none of the existing approaches support the simultaneous optimization of multiple objectives while considering the battery constraint and generating a whole Pareto-set of EV travel plans combining routes and recharging stops on country-sized road networks. Consequently, existing approaches make it difficult to properly address the trade-offs EV users can have between travel planning objectives and, in particular, properly incorporating increasingly important pricing considerations into planning EV trips. A summary of the most relevant approaches and their differences can be seen in Table 1.

## Multi-objective EV travel planning problem

We model the EV travel planning problem as a multi-objective constrained shortest path problem with SoC constraints and charging stops with two optimization objectives: time and cost. Formally, we define the EV travel planning problem as a tuple $\langle W, R\rangle$ where $W$ is the global static *EV travel planning environment* and $R$ is the *EV travel planning request* that is specific for each EV user and their needs. The solution to an EV travel planning problem is the Pareto-set of EV travel plans $\Pi$. The structure of the EV travel planning problem is visualized in Figure 2. We describe each part in a dedicated section below.

### EV travel planning environment

The EV travel planning environment (termed *planning environment* further on) represents the road network and charging stations, i.e. the components of the travel planning problem that are independent of the specific details of individual planning requests.

The planning environment is a tuple $W = \langle G, Q\rangle$, where $G = \langle V, E, \tau, d\rangle$ is a weighted oriented graph representing the underlying road network, with $V$ being the set of graph nodes representing intersections and $E$ the set of graph edges representing road segments. Each edge $e \in E$ has a defined traversal duration $\tau(e) \in \mathbb{R}^+$ and a length $d(e) \in \mathbb{R}^+$. Although the algorithm does not require it, we assume for the sake of simplicity that the graph $G$ is without loops and parallel edges.

The set of charging stations $Q$ defines the locations where EVs can be charged. Each charging station $q \in Q$ is defined as

**Table 1.** Summary of key properties of the most relevant approaches.

| | multi-objective | cost objective | country-scale road network | realistic # of CS | optimal |
|---|---|---|---|---|---|
| Baum et al. (2019a) | × | × | ✓ | ✓ | ✓ |
| Baum et al. (2019b) | × | × | ✓ | ✓ | ✓ |
| Storandt (2012) | × | × | ✓ | × | ✓ |
| Ben Abbes et al. (2022) | ✓ | ✓ | × | × | × |
| Siddiqi et al. (2011) | ✓ | ✓ | × | × | × |
| Schoenberg and Dressler (2023) | ✓ | × | ✓ | × | ✓ |

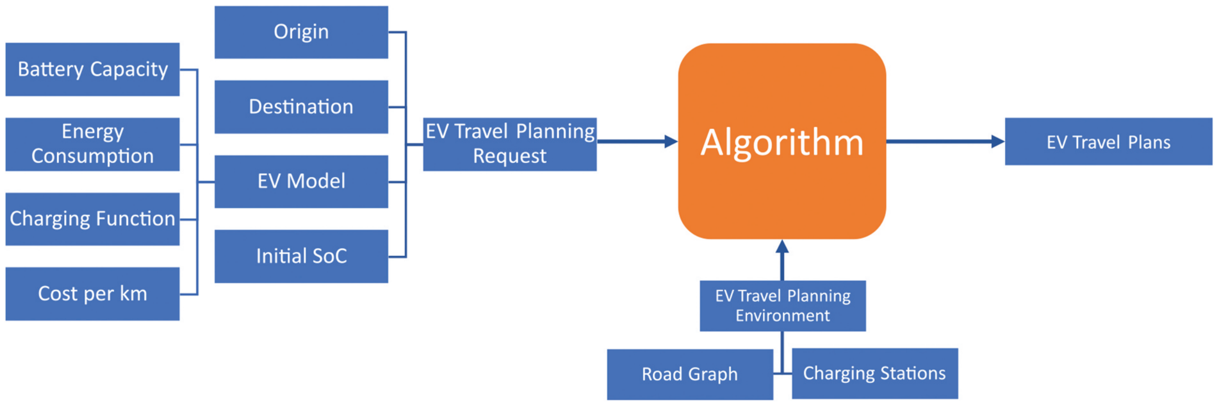


**Figure 2.** Schema of the EV travel planning problem.

a tuple $q = \langle v_q, P_q, \gamma_q \rangle$, where $v_q \in V$ is the node where the charging station is located, $P_q \in \mathbb{R}^+$ is the maximum power the charging station provides (*charging rate*) and $\gamma_q : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}_0^+$ is the *charging cost function* that defines how much any charging session at the station $q$ costs based on the duration $t \in \mathbb{R}^+$ of the session and the amount of energy $j \in \mathbb{R}^+$ charged during the session.

The *charging cost function* can formalize various types of charging policies, including all of those popular today, such as fixed price per charging session, price per minute of charging, price per kWh of charged energy, or their combination. Since each charging station $q$ may have defined its own cost function $\gamma_q$, our proposed formalism also allows modeling the *location-of-use* pricing, where the price of charging depends on the location of the charging station. In fact, the algorithmic approach we propose to solve the EV travel planning problem can work with even more complex types of arbitrary non-negative pricing policies, in particular, with the *time-of-use* pricing, where cost functions $\gamma_q$ depend on the time of pricing. Although totally compatible with our approach, we do not further consider *time-dependent* pricing functions, primarily for the sake of simpler presentation and also because there are not yet enough real-world data that could be used for the evaluation of EV travel planning with time-dependent charging prices. That said, we still support different prices and charging rates of individual charging stations.

### *EV travel planning request*

The EV travel planning request defines the user's specific request for EV travel planning. The request is defined as a tuple $R = \langle v_{\text{init}}, v_{\text{goal}}, b_{\text{init}}, M \rangle$, where $v_{\text{init}} \in V$ is the origin, $v_{\text{goal}} \in V$ is the destination, $b_{\text{init}} \in [0, b_{\max}]$ is the initial SoC and $M$ is the EV model the user drives (see the following section).

### *EV model*

The electric vehicle model mainly defines how the energy in the battery is consumed while driving and how much time it takes to recharge the energy at charging stations while taking into account the current SoC. The electric vehicle model $M = \langle b_{\max}, \beta, \phi, \psi \rangle$ consists of the *maximum battery capacity* $b_{\max} \in \mathbb{R}^+$ of the EV, *cost per km of driving* $\psi \in \mathbb{R}_0^+$, and two functions defining how the EV consumes the energy stored in its battery and how the battery is recharged.

The *energy consumption function* $\beta : E \to \mathbb{R}$ defines the amount of energy the EV spends traversing edge $e \in E$. The energy consumption function can take into account various properties of the edge, such as the length, speed, or elevation profile. Note that the consumption can be negative due to recuperation.

The *charging function* $\phi : [0, b_{\max}] \times (0, b_{\max}] \times \mathbb{R}^+ \to \mathbb{R}^+$ defines the time needed to complete a charging session specified by the starting SoC $b_{\text{start}} \in [0, b_{\max}]$, the final SoC $b_{\text{end}} \in (0, b_{\max}]$ and the maximum available power $P_{\max} \in \mathbb{R}^+$. An example of a simplified representation of a piecewise linear charging function is given in Figure 3. As already mentioned, the time required to charge EVs usually differs significantly based on the starting SoC. For example, it is usually much slower to charge the battery from 80% to 100% of battery capacity than from 20% to 40% of battery capacity. The charging function models this behavior.

The *cost per km of driving* $\psi \in \mathbb{R}_0^+$ defines EV wear-and-tear costs per driven distance. We introduced it to explicitly account for wear-and-tear costs as this provides a more accurate model of the real-word optimization problem faced by the EV drivers. Moreover, it helps to avoid unreasonably long detours to free charging stations. Without the driving cost, the most cost-efficient route would be a route through a series of free charging stations even though it would require absurd detours.

### *EV travel plan*

The *EV travel plan* for a vehicle model $M = \langle b_{\max}, \beta, \phi, \psi \rangle$ in a planning environment $W = \langle G, Q \rangle$ is a sequence of interleaving states and actions $\pi = (s_0, a_0, s_1, a_1, \ldots, a_{k-1}, s_k)$.

A *state* $s_i$ fully describes the status of the EV and the value of plan objectives at the i-th step of the plan and action $a_i$ describes the transition between the states $s_i$ and $s_{i+1}$.

We define the state $s$ as a tuple $\langle v, t, c, b \rangle$ where:

- $v \in V$ is an EV location graph node.
- $t \in \mathbb{R}_0^+$ is the time at which the state is reached.
- $c \in \mathbb{R}_0^+$ is the charging and driving cost spent to reach the state.

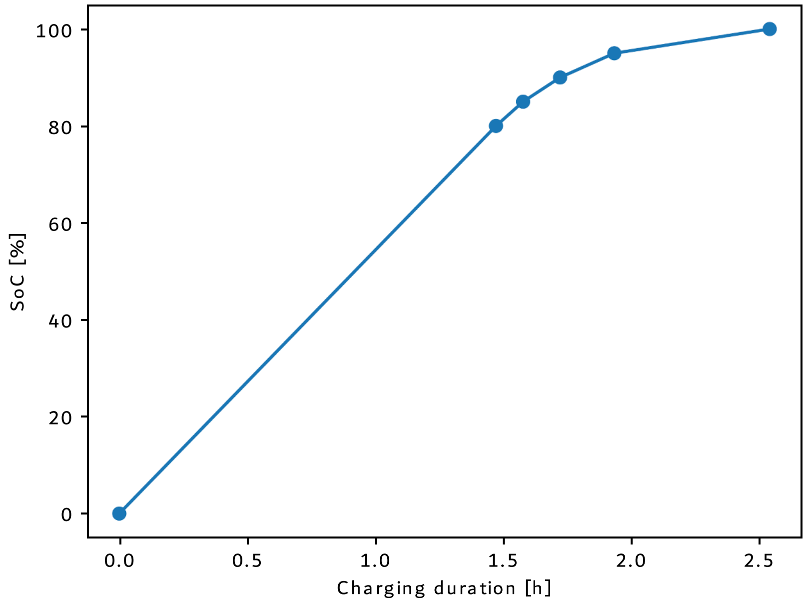


**Figure 3.** Piecewise linear charging function for an EV with 40kWh battery on 20kW charging station. It is a simplified representation depicting the charging duration starting from the empty battery. The duration of charging starting from a different SoC can be calculated as a difference between the duration of charging from an empty battery to the target SoC and the duration of charging from an empty battery to the starting SoC.

- $b \in [0, b_{\max}]$ is the SoC with which the state is reached (higher value means more energy in the battery).

An EV travel plan consists of two types of *actions*:

- *move*$(e)$ that moves the vehicle across the edge $e = (v, u) \in E$:
  $$\langle v, t, c, b\rangle \longrightarrow \langle u, t + \tau(e), c + \psi d(e), \min(b - \beta(e), b_{\max})\rangle$$
- *charge*$(q, j)$ that charges the vehicle at the charging station $q \in Q$ with energy $j \in \mathbb{R}^+$:
  $$\langle v_q, t, c, b\rangle \longrightarrow \langle v_q, t + t_{q,j}, c + \gamma_q(t_{q,j}, j), b + j\rangle$$
  where $t_{q,j} = \phi(b, b + j, P_q)$

Since a state fully describes the EV status and objective values, the last state of a plan can also be used as a simpler representation of the plan. For example, the state $s = \langle v, 950\text{s}, 5, 10\text{kWh}\rangle$, represents a plan that takes 950 seconds, costs 5€ and the EV ends at node $v$ with 10kWh of energy in the battery.

In order for the EV travel plan $\pi = (s_0, a_0, s_1, a_1, \ldots, a_{k-1}, s_k)$ to be valid, the state of charge must not drop below zero or get above the maximum battery capacity $b_{\max}$: $0 \le b_i \le b_{\max}, \quad i \in 0, \ldots, k$.

We say that an EV travel plan $\pi$ with $k + 1$ states is feasible for a planning request $R = \langle v_{\text{init}}, v_{\text{goal}}, b_{\text{init}}, M\rangle$ if it is valid, $v_0 = v_{\text{init}}, b_0 = b_{\text{init}}$ and $v_k = v_{\text{goal}}$. We also define the *plan time* as $t_\pi = t_k$ and the *plan cost* as $c_\pi = c_k$.

### EV travel plan objectives and dominance

An EV travel planning algorithm should produce EV travel plans feasible for planning request $R$ optimal with regard to two objectives – time and cost. More specifically, the goal of the algorithm is to minimize $t_\pi$ and $c_\pi$.

Since there is more than one optimization objective, a total ordering with regard to $t_\pi$ and $c_\pi$ does not usually exist. However, a partial ordering exists according to weak dominance:

**Definition 1.** Let $\pi, \pi'$ be two EV travel plans. We say that $\pi$ *weakly dominates* $\pi'$ (denoted as $\pi \preceq \pi'$) iff $t_\pi \le t_{\pi'}$ and $c_\pi \le c_{\pi'}$.

Further, we refer to the *weak dominance* only as the *dominance* for simplicity.

Unfortunately, it is not always the case that either of the two plans dominates the other (neither $\pi \preceq \pi'$ nor $\pi' \preceq \pi$ is true). For example, plan A takes 2 hours and costs 10€, while plan B takes 90 minutes and costs 20€. We cannot simply order the plans by their cost and duration. Plan A is cheaper, but plan B is faster. There is a trade-off between the two objectives. In such cases, we say that the two plans are *non-dominated*.

### EV travel planning problem solution

The solution to the multi-objective EV travel planning problem defined by a planning request $R$ and a planning environment $W$ is a set of feasible Pareto-optimal (non-dominated) EV travel plans $\Pi$. The travel plans are optimal regarding the total travel time $t_\pi$ and the cost $c_\pi$ minimization objectives. The EV travel plans in the resulting Pareto-set $\Pi$ express the possible trade-offs between the two objectives – the travel and charging time $t_\pi$ and the charging and driving cost $c_\pi$.

### Problem definition summary

Given a road graph $G$, set of charging stations $Q$ with charging cost functions, EV model $M$ with charging and consumption functions, and a planning request $R = \langle v_{\text{init}}, v_{\text{goal}}, b_{\text{init}}, M\rangle$ defined by the origin $v_{\text{init}}$, destination $v_{\text{goal}}$ and initial SoC $b_{\text{init}}$, the goal is to find a Pareto-optimal set of plans (consisting of move and charge actions) that minimizes their time and cost. The time objective comprises travel time and time spent by charging, and the cost objective comprises the wear-and-tear costs defined by *cost per km* and the cost of charging.

## Multi-objective EV travel planning algorithm

To solve the above-outlined EV travel planning problem, we have developed two algorithms based on the concept of constrained multi-objective shortest path search that leverages several techniques proposed for related problems to our new problem.

The two algorithms – optimal and approximate – enable to manage the trade-off between the solution quality/optimality and the performance.

Both versions of our EV travel planning algorithms rely on an informed constrained multi-objective search that is based on a constrained extension of the multi-objective A* algorithm (Mandow et al. 2005), which is capable of solving similar problems on country-sized graphs with millions of nodes (Baum, Dibbelt, Gemsa, et al. 2019; Schoenberg and Dressler 2023) in contrast to the meta-heuristic approach that can solve the problem only on tiny graphs with at most hundreds of nodes (Ben Abbes, Rejeb, and Baati 2022). The core idea of A*-based algorithms is the heuristics estimating the objectives at the destination that are used to guide the search and to prune sub-optimal plans earlier.

To further improve the response time of the algorithm, we present an approximate version of our EV travel planning algorithm utilizing the dominance relaxation speed-up technique.

### *States and their dominance*

To describe the algorithm, we use the same definition of states $s = \langle v, t, c, b \rangle$ as presented in the problem definition.[6] As mentioned above, a state can also be viewed as a simpler representation of a (partial) EV travel plan since it fully describes all essential attributes that are necessary for the planning algorithm to decide about the subsequent actions. We say that a plan is *partial* if its last location is not the destination.

In this section, we formally extend the concept of EV travel plan *dominance* (Definition 1) to states while maintaining full compatibility. The algorithm requires two versions of the dominance that are used in different algorithm steps. $\pi$-dominance in Definition 2 is a straightforward adjustment of Definition 1 to the context of states leveraging the information provided by the time and cost heuristics $h_t$ and $h_c$. The algorithm uses $\pi$-dominance when it checks the explored states against the already found solution plans.

**Definition 2.** Let $s = \langle v, t, c, b \rangle, s' = \langle v', t', c', b' \rangle$ be two states. We say that $s$ $\pi$-dominates $s'$ (denoted as $s \preceq_\pi s'$) iff the following conditions are satisfied:

$$\begin{aligned} t &\leq t' + h_t(s') \\ c &\leq c' + h_c(s') \end{aligned} \tag{1}$$

However, $\pi$-dominance does not work if both states represent partial plans (not at the destination yet). For example, a state representing a partial plan that is slower but has a higher SoC could lead to a faster plan at the destination because it could have enough energy to reach the destination without any additional stop at a charging station. Therefore, the algorithm requires the following dominance extended by the SoC attribute and without the heuristic estimates to check the states representing partial plans (details how it is used in the section below).

**Definition 3.** Let $s = \langle v, t, c, b \rangle, s' = \langle v', t', c', b' \rangle$ be two states at the same node ($v = v'$). We say that $s$ dominates $s'$ (denoted as $s \preceq s'$) iff all the following conditions are satisfied:

$$\begin{aligned} t &\leq t' \\ c &\leq c' \\ b &\geq b' \end{aligned} \tag{2}$$

At last, we introduce the dominance between a state and a set of non-dominated states.

**Definition 4.** Let $s$ be a state and $S$ be a set of mutually non-dominated states according to the dominance relation We say that $S$ dominates $s$ (denoted as $S \preceq s$) iff

$$\exists s' \in S : s' \preceq s \tag{3}$$

### *Optimal EV travel planning algorithm*

As mentioned above, the optimal algorithm is based on a multi-objective version of A* algorithm guiding the search by two heuristics. We designed remaining travel time heuristic $h_t$ and minimum remaining charging and driving cost heuristic $h_c$.

To further reduce planning times without sacrificing optimality, we employed a technique that significantly reduces the computational complexity of the dominance checks, which are the greatest bottleneck of the proposed algorithm. The *dimensionality reduction* technique (described below) allows to significantly reduce the size of the Pareto-sets maintained during the search.

The proposed optimal algorithm finds the whole Pareto-optimal set of travel plans (see Appendix A.1 for proof).

The pseudocode of the optimal algorithm is given in Algorithm 1. The algorithm uses four basic types of data structures:

- Pareto-set of visited/closed states $S_v^{\mathrm{cl}}$ for each graph node $v \in V$ that contains all states that were already visited and expanded by the algorithm.[7]
- Pareto-set of opened states $S_v^{\mathrm{op}}$ for each graph node $v \in V$ that holds the states that were generated but not yet visited by the algorithm.
- Solution set $\Pi$ with the states representing plans that reached the destination.
- Set of all opened states $S^{\mathrm{op}} = \bigcup_{v \in V} S_v^{\mathrm{op}}$, that can also be viewed as a priority queue for the states to be visited.

In each iteration, a lexicographically minimal state $s_{\min}$ is extracted from the set of all opened states $S^{\mathrm{op}}$ (*extractMin* on line 11). The states $s = \langle v, t, c, b \rangle$ are sorted first by their estimated time $t + h_t(s)$ and then by cost $c + h_c(s)$ and SoC $b$.

Each extracted state is first checked for whether it is not $\pi$-dominated by any of the already found solution states (line 12) and whether it is not a solution itself (*inDestination* on line 14). If neither

**Algorithm 1**: Pseudocode of the optimal multi-objective EV travel planning algorithm.

**Input:** planning environment $W = \langle G, Q \rangle$
planning request $R = \langle v_{\mathsf{init}}, v_{\mathsf{goal}}, b_{\mathsf{init}}, M \rangle$
EV model $M = \langle b_{\mathsf{max}}, \beta, \phi, \psi \rangle$
**Output:** set of Pareto-optimal travel plans $\Pi$

1 **function** Plan
2 $S_v^{\mathsf{op}}$: set of opened states for each graph node $v \in V$
3 $S_v^{\mathsf{cl}}$: set of visited/closed states for each graph node $v \in V$
4 $S^{\mathsf{op}} = \bigcup_{v \in V} S_v^{\mathsf{op}}$: set of all opened states
5 $\Pi$: set of solution states
6 $S_v^{\mathsf{op}} \leftarrow \emptyset, \forall v \in V$
7 $S_v^{\mathsf{cl}} \leftarrow \emptyset, \forall v \in V$
8 $\Pi \leftarrow \emptyset$
9 $S_{v_{\mathsf{init}}}^{\mathsf{op}} \leftarrow \{\langle v_{\mathsf{init}}, 0, 0, b_{\mathsf{init}} \rangle\}$
10 **while** $S^{\mathsf{op}} \neq \emptyset$ **do**
11 $s_{\min} \leftarrow$ `extractMin`$(S^{\mathsf{op}})$
12 **if** $\Pi \preceq_\pi s_{\min}$ **then**
13 **continue**
14 **if** `inDestination`$(s_{\min})$ **then**
15 $\Pi \leftarrow \Pi \cup \{s_{\min}\}$
16 **else**
17 $S_{v_{\min}}^{\mathsf{cl}} \leftarrow S_{v_{\min}}^{\mathsf{cl}} \cup \{s_{\min}\}$
18 $S \leftarrow$ `expand`$(s_{\min})$
19 **forall** $s = \langle v, t, c, b \rangle \in S$ **do**
20 **if** $b < 0$ **then**
21 **continue**
22 **if** $(S_v^{\mathsf{op}} \cup S_v^{\mathsf{cl}}) \preceq s \vee \Pi \preceq_\pi s$ **then**
23 **continue**
24 **else**
25 $S_v^{\mathsf{op}} \leftarrow S_v^{\mathsf{op}} \setminus \{s' \in S_v^{\mathsf{op}} | s' \preceq s\}$
26 $S_v^{\mathsf{op}} \leftarrow S_v^{\mathsf{op}} \cup \{s\}$
27 **return** $\Pi$

of the conditions is true, the state is added to the corresponding visited/closed set $S_v^{\mathrm{cl}}$ (line 17) and expanded (*expand* on line 18).

Let $s_{\min} = \langle v, t, c, b \rangle$ be the extracted state. The state is then expanded (function) using the following actions corresponding to the actions described in the EV travel plan definition:

(i) **move** For each outgoing edge $e = (v, u) \in E$, a new state

$$s = \langle u, t + \tau(e), c + \psi d(e), \min(b - \beta(e), b_{\max}) \rangle$$

is generated.

(ii) **charge** For each charging station $q = \langle v_q, P_q, \gamma_q \rangle$ such that $v_q = v$ and for each amount of energy $j$ from a predefined set of target charging levels (for example, charging to 80%, 90%, 100% of battery capacity) a new state

$$s = \langle v_q, t + t_q, c + \gamma_q(t_q, j), b + j \rangle$$

where $t_q = \phi(b, b + j, P_q)$ is generated. The predefined set of target charging levels can be configured arbitrarily, but it should take into account the shape of the charging function $\phi$. For example, if the function is piecewise linear, it should include the breakpoints.

We use the discretization of the target charging levels to significantly reduce the number of newly generated states.[8]

All the newly generated states are first checked if they violate the SoC constraint. If they do, they are pruned immediately (line 21). Then, they are checked if they are not dominated by any of the states in their corresponding $S_v^{\mathrm{op}}$ and $S_v^{\mathrm{cl}}$ Pareto-sets (line 22). Additionally, they are also checked if they are not $\pi$-dominated by any of the already found solution states in $\Pi$. If they are not dominated, they are added to the opened set $S_v^{\mathrm{op}}$ while removing all states in the opened set dominated by the newly generated one (lines 25–26).

### *Remaining travel time heuristic*

This heuristic relaxes the battery constraints and estimates the minimum time needed to reach the destination regardless of the battery constraints. It calculates a lower bound on the travel time to the destination.

Let $s = \langle v, t, c, b \rangle$ be a state, then the heuristic can be expressed as

$$h_t(s) = t(v, v_{\mathrm{goal}}) \tag{4}$$

where $t(v, v_{\mathrm{goal}})$ is the minimum travel time needed to drive from $v$ to $v_{\mathrm{goal}}$.

The heuristic can be pre-calculated (see Appendix C for details).

### *Minimum remaining charging and driving cost heuristic*

Since the cost objective comprises of two components – the charging cost and the driving cost – the heuristic is based on the combination of the lower bounds of both individual components. The calculation of the minimum cost spent on charging is based on the most energy-efficient route to the destination while the minimum driving cost is based on the length of the shortest route.

Let $s = \langle v, t, c, b \rangle$ be a state, then the heuristic can be expressed as

$$h_c(s) = b_{\min} c_{\min} + \psi d(v, v_{\mathrm{goal}}) \tag{5}$$

where $b_{\min}$ is the minimum amount of energy that has to be charged to reach the destination (details below), $c_{\min}$ is the minimum possible price per amount of energy achievable with regards to the cost functions of all charging stations and the charging function of the EV (details in Appendix B.2), and where $d(v, v_{\mathrm{goal}})$ is the length of the shortest path from $v$ to $v_{\mathrm{goal}}$.

The minimum amount of energy that has to be charged to reach the destination $b_{\min} = \beta(v, v_{\mathrm{goal}}) - b$ is the amount of energy required by the most energy efficient route from $v$ to $v_{\mathrm{goal}}$ deducted by the current SoC $b$.

The heuristic can be pre-calculated (see Appendix C for details).

### *Dimensionality reduction*

The greatest bottleneck of our proposed algorithm is the computational complexity of dominance checks that is directly dependent on the size of the Pareto-sets managed by the algorithm ($S_v^{\mathrm{op}}$, $S_v^{\mathrm{cl}}$, and $\Pi$). The size of the Pareto-sets can grow exponentially with the size of the problem (in particular, with the size of the road graph and the number of charging stations) and the number of components on which the dominance is based, making the dominance checks very expensive.

Fortunately, we can leverage a technique proposed by Pulido et al. (2015) that reduces the dimension of some of the Pareto-sets without the loss of optimality. If we use the lexicographical ordering for the minimal label $s_{\min}$ extraction (line 11 in Algorithm 1) and if the heuristic estimates $h_t$ and $h_c$ are consistent, we can remove the first attribute (in our case the time) from the dominance checks against the solution set $\Pi$ (line 12) and against the closed set $S_v^{\mathrm{cl}}$ (line 22). Unfortunately, it does not apply to the opened set $S_v^{\mathrm{op}}$.

The core idea is that if the algorithm extracts the best labels at first by time (lexicographical ordering) and if the heuristics are consistent, we know that the closed and solution sets cannot contain labels that are worse in the time attribute than the labels extracted later. Pulido et al. (2015) prove the correctness of the technique if the requirements are satisfied.

### Approximate EV travel planning algorithm

Unfortunately, the optimal version of our EV travel planning algorithm even with dimensionality reduction is too slow (see the Evaluation results section). Therefore, to achieve practically usable planning times, we employed *ϵ-dominance relaxation* (Batista et al. 2011) of dominance conditions. For example, dominance from Definition 3 is extended to:

**Definition 5.** Let $s = \langle v, t, c, b\rangle, s' = \langle v', t', c', b'\rangle$ be two states and $\varepsilon_t, \varepsilon_c, \varepsilon_b \in [0, 1]$ be relaxation ratios. We say that $s$ *ε-dominates* $s'$ (denoted as $s \preceq_\varepsilon s'$) iff the following conditions are satisfied:

$$\begin{array}{ccc} \varepsilon_t \cdot t & \leq & t' \\ \varepsilon_c \cdot c & \leq & c' \\ b & \geq & \varepsilon_b \cdot b' \end{array} \tag{6}$$

All variants of dominance defined above can be adapted in a similar fashion. This technique is compatible with both heuristics and also with dimensionality reduction.

This relaxation speeds up the algorithm by pruning more states during the search; however, it does not maintain optimality. Therefore, the ratios need to be selected carefully to achieve a good trade-off between the reduction of the planning time and the loss of solution quality.

## Evaluation setup

In this section, we describe in detail variants and configurations of the proposed algorithm we evaluated together with used instances of the EV Travel Planning Problem. We also describe evaluation metrics.

### Evaluation problem instances

The EV planning environments used for the evaluation were constructed from real-world data sets for Germany. Germany has a large road network with many charging stations and good accessibility of data. Besides the large-scale Germany area, we also performed the evaluation on a smaller-scale area of the German state of Bavaria (visualization of the areas can be seen in Figure 4). The Bavaria area was used because some of the evaluated algorithm variants could not solve instances for the whole Germany in a reasonable time.

We extracted road *graphs* for both Germany (1.5M nodes) and Bavaria (300k nodes) from OpenStreetMaps[9] (excluding local, residential roads) and then mapped real-world charging stations[10] to them (visualization can be seen in Figure 4). The elevation data were gathered from SRTM.[11]

The *charging station* datasetftn:dataset-cs contains 12,633 charging stations (2,225 in Bavaria). Each charging station is described by its location (GPS), the maximum power (kW), and the pricing policy. The pricing policies are of four types: energy-based, duration-based, fixed, and occupancy-based (details in Appendix D.1). The pricing policies vary a lot between charging stations, implementing the so-called *location-of-use* pricing. We also created a set of charging stations, where all free charging stations are replaced with paid ones to evaluate the impact of free charging stations on the planning time.[12]

We model the *energy consumption* of the EV with a linear model that takes into account the length and the elevation profile of the roads similarly to Eisner et al. (2011). We set the model to approx. correspond to 250 km range with 40 kWh *battery capacity* and to 500 km range with 80 kWh battery. We used a piecewise linear *charging function* similar to Baum et al. (2019b) that expresses well the decreasing charging speed when the state of charge approaches the maximum battery capacity while maintaining simplicity. The charging speed gradually decreases starting at 80% battery capacity with other breakpoints at 85%, 90%, and 95%. The charging speed is 6.6× slower while above 95% than below 80%. The *cost per km* is set to 0, 3, or 10 cents per km.

We generated the planning requests as 1,000 random *origin-destination* pairs, uniformly sampled from road graph nodes, for each evaluation area (i.e. 2,000 in total). The origin-destination pairs were generated so that the direct distance was at least 250 km. The initial SoC was set to 100% of the battery capacity.

Additional details can be seen in Appendix D. We will provide all the datasets on request.

### Evaluated algorithm configurations

We have proposed two variants of the EV travel planning algorithm: optimal and approximate.

We first evaluate the impact of the individual components of the *optimal* variant of the algorithm (heuristics and dimensionality reduction). Then, we evaluate the *approximate* variant of the algorithm employing ε-relaxation. The evaluation compares the various ε-relaxation configurations to the optimal variant in terms of planning time and solution quality (we discuss the Evaluation metrics section). Specifically, we evaluated each epsilon coefficient – time, cost, and SoC $(\varepsilon_t, \varepsilon_c, \varepsilon_b)$ – individually while the remaining coefficients were set to 1.0 and we also evaluated all coefficients combined (all set to the same value). The examined coefficients are: $\{0.995, 0.99, 0.98, 0.96, 0.93, 0.9\}$. Altogether in this paper, we evaluated 24 different ε-dominance configurations of the algorithm. Note that our goal in the evaluation was to evaluate the influence of the individual epsilon coefficients on the behavior of the algorithm rather than trying to find a single best possible configuration. In fact, there is not a single best configuration due to the trade-off between the algorithm speed and solution quality and final selection of the coefficients can be considered as fine-tuning that is highly dependent on the specific use case and environment.

The algorithm is configured to generate new states for charge action to the following target charging levels (details in the algorithm description section) based mostly on the used charging function breakpoints: $\{10\%, 20\%, 30\%, \ldots, 80\%, 85\%, 90\%, 95\%, 100\%\}$.

### Evaluation metrics

Since most variants and configurations of the algorithm are not capable of finding the complete solution for more complex problem instances in a reasonable time, we introduced a 2 h time limit after

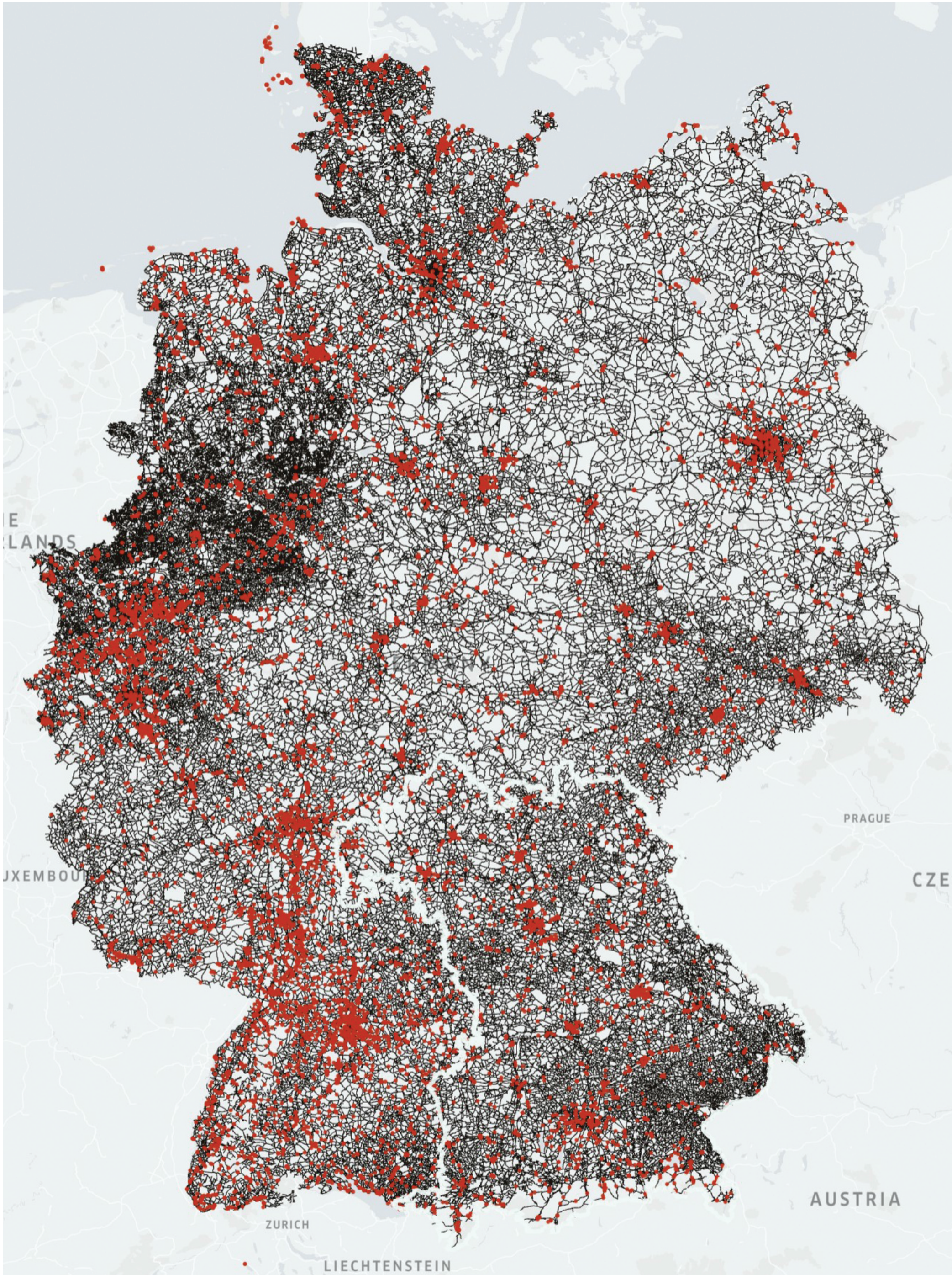


**Figure 4.** Germany road graph with charging stations. Bavaria state outlined bottom right.

which the planning is terminated. Therefore, we use the *completion percentage* which we define as the percentage of instances for which the algorithm found the complete solution Pareto-set $\Pi$ within the time limit, as the primary performance metric.

Additionally, we use the *planning time speed-up* as the secondary metric. We define it as $t^{base}_{run}/t_{run}$, where $t_{run}$ and $t^{base}_{run}$ are the *planning times* for the measured algorithm configuration and the baseline configuration, respectively.

Because the $\varepsilon$-relaxation technique does not preserve optimality, we need to measure also its impact on the *solution quality loss*. We measure the *solution quality loss* as the closeness of the resulting set of EV travel plans to the optimal Pareto-set of plans. From many multi-objective solution quality metrics, surveyed by Laszczyk and Myszkowski (2019) and Riquelme et al. (2015), the best fit for our case is the *average distance* metric proposed by Hrnčíř et al. (2016):

$$d(\Pi^*, \Pi) = \frac{1}{|\Pi^*|} \sum_{\pi^* \in \Pi^*} \min_{\pi \in \Pi} d(\pi^*, \pi)$$

The average distance of the Pareto-set $\Pi$ from the full Pareto-set $\Pi^*$ measures the average Euclidean distance in the objective space (in our case time and money cost) normalized to $[0, 1]$ range. For each objective, the minimum value from all plans $\Pi \cup \Pi^*$ is mapped to 0, and correspondingly, the maximum value is mapped to 1. For illustration, if the optimality loss was 7% equally distributed among the objectives, the distance $d(\pi^*, \pi)$ would be approx. 0.1.

### HW and SW details

We implemented our EV travel planning algorithms in Java. We ran the experiments on the OpenJDK 64-Bit Server VM Temurin-17.0.4 JVM on a computing cluster node with 64 cores/128 threads 3.1 GHz (2 × AMD EPYC 7543). We ran multiple instances simultaneously while limiting the resources to 8 threads and 31GB of RAM per instance.

In summary, we ran experiments for 780 combinations of planning environments, EV models, and algorithm configurations. For each combination, solutions for 1,000 planning requests were calculated resulting in a total of 780,000 problem instances calculated. The CPU time needed to do the computations was more than 2 million CPU hours.

## Evaluation results

We conducted a thorough evaluation of the proposed EV travel planning algorithms on real-world data (described above). We first focus on evaluating the *optimal algorithm* and the impact of various EV Travel Planning Problem parameters on its planning time. Then, we focus on the *approximate algorithm* and the trade-off between planning time and the quality of resulting travel plans.

### Evaluation results: optimal algorithm

First, we evaluated the impact of the components of the optimal algorithm, i.e. dimensionality reduction and the time and cost heuristics. Then, we evaluated the impact of problem parameters on the planning time of the optimal algorithm. The problem and algorithm configuration parameters in the evaluation can be seen in Table 2.

**Table 2.** Overview of used problem and algorithm parameters used for the evaluation of *optimal algorithm*.

| Problem Parameters | |
|---|---|
| Environment | **Germany**, Bavaria |
| CS pricing policy | **all**, paid-only |
| EV range [km] | **250**, 500 |
| $\psi$ [€/km] | 0, **0.03**, 0.10 |
| **Algorithm Parameters** | |
| Dimensionality reduction | **on**, off |
| Time heuristic $h_t$ | **on**, off |
| Cost heuristic $h_c$ | **on**, off |
| $\varepsilon_t$ | **1.0** |
| $\varepsilon_c$ | **1.0** |
| $\varepsilon_b$ | **1.0** |

Note: Default parameters used unless stated otherwise are bold.

### Evaluating the impact of algorithm parameters

As mentioned above, the optimal algorithm comprises three components (dimensionality reduction and the time and cost heuristics), and each of them can be used independently, i.e. turned on/off. We compared various combinations of the components turned on (e.g. only cost heuristic with dimensionality reduction) to the combination that uses all components, i.e. the full optimal algorithm. Unless stated otherwise, the following planning environment and EV model are used: the whole Germany graph with free charging stations, EV range 250 km, and cost per km set to 0.03€.

Figure 5 shows that all individual components are crucial for the optimal algorithm to maintain reasonable performance on country-size graphs (Germany) at least for experimental purposes. Without any of the heuristics, the algorithm would be basically useless, and without dimensionality reduction, the algorithm is capable of solving only 50% of the instances. On smaller planning environments (Bavaria), the algorithm with at least two used components solves a great majority of instances, and even on this much smaller planning environment, the average speed-up of cost heuristic is ∼150×, of time heuristic ∼70× and of the dimensionality reduction ∼4×.[13] The results also show that dimensionality reduction dramatically reduces the size of stored Pareto-sets and significantly reduces memory requirements since there is a minimum amount of instances terminated because of insufficient memory (purple bars).

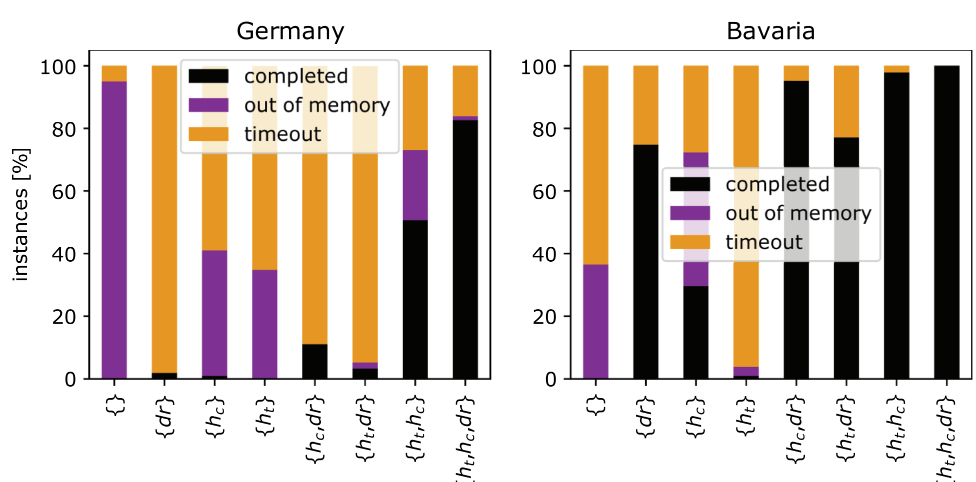


**Figure 5.** Completion percentage for different combinations of optimal algorithm components. $h_t$ – time heuristic, $h_c$ – cost heuristic, $dr$ – dimensionality reduction.

Although the speed-up of the heuristics and the dimensionality reduction is very significant, the planning time is still not sufficient for solving more complex country-size instances in a real-world application. While the optimal algorithm performs quite well on smaller areas (avg. planning time is 74 seconds on Bavaria), the average planning time on Germany planning environment is more than 18 minutes.

**Table 3.** The number of completed instances for different costs per km ($\psi$).

| $\psi$[€/km] | completion [%] |
|---|---|
| 0 | 98.3 |
| 0.03 | 82.6 |
| 0.10 | 82.7 |

### *Evaluating the impact of problem parameters*

We also evaluated the impact of the problem parameters on the planning time of the optimal algorithm. We examined the influence of the availability of free charging, the cost per km of driving, the EV range, and the origin-destination distance.

Contrary to our expectations, free charging stations reduce the planning time. Since the lower bound of the charging part of the cost heuristic is always zero with free charging stations and therefore less informed, we expected it would be a significant performance hit. While the optimal algorithm is capable of successfully solving 92.6% of the planning requests with free charging stations, only paid charging stations reduce the percentage to 33%. The average speed-up with free charging stations is $\sim 11\times$ on completed instances caused by a significant reduction of the average Pareto-set size from 500 with paid-only charging stations to 300 with free charging stations.

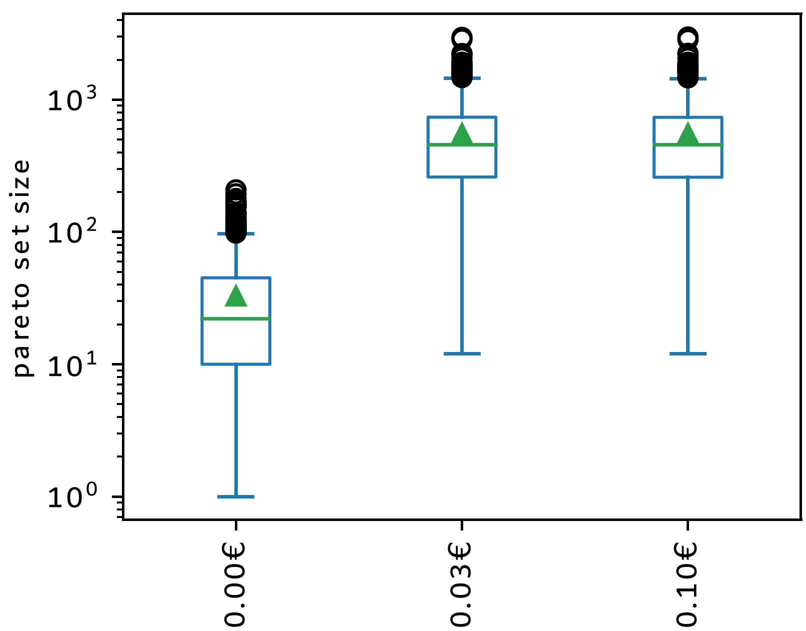


**Figure 6.** Distribution boxplots[14] of the sizes of travel plan pareto-sets for different costs per km.

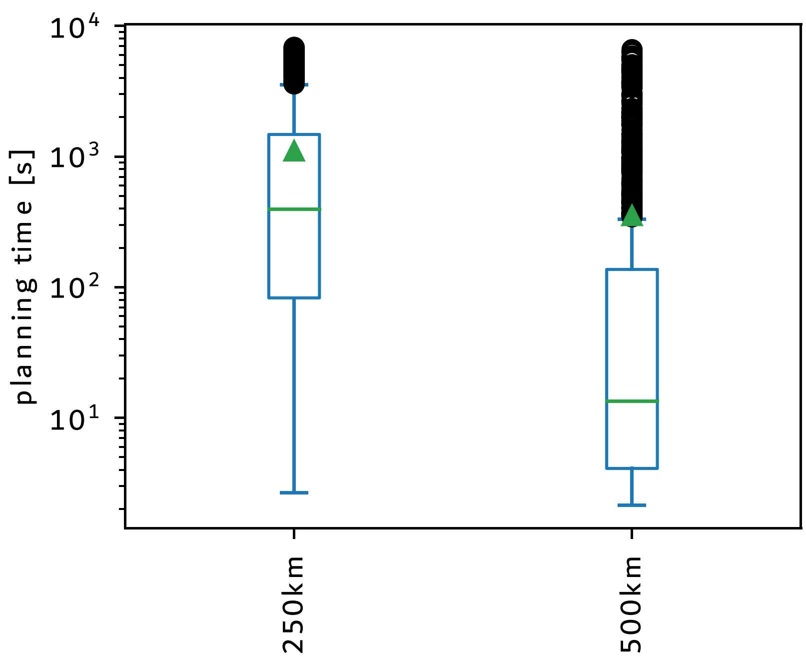


**Figure 7.** Distribution boxplot of planning times for different EV ranges. The green triangles are the means, and the green lines are the medians.

The introduction of cost per km of driving leads to increased complexity. As you can see in Table 3, zero driving costs allow the algorithm to solve nearly all instances, while the inclusion of cost per km reduces the completion percentage to 83%.

The root cause is the dramatically increased sizes of solution Pareto-sets. Figure 6 shows that the average Pareto-set size is increased more than 16 times from 33 to 553 for 0.03€/km. It leads to ∼2.5× increased average planning time from 480s to 1105s for 0.03€/km. Such a large solution size growth should lead to a much smaller number of completed instances; however, it is largely mitigated by the cost heuristic. The solution size growth is caused by the trade-off of time and distance that is introduced into the cost objective by the cost per km parameter.

Another important parameter of problem instances is the EV range. A greater range leads to a greater completion – 82.6% for 250 km range vs. 97.6% for 500 km range. More interesting is the great difference in average planning times (1105s vs. 354s) and even greater in median planning times (395s vs. 13s) – see Figure 7.

The large planning time drop is caused by a smaller number of required charging sessions to complete the trip. As you can see in Figure 8, over 50% of instances with 500 km range EV do not require any charging while some instances with 250 km EV range require even more than 2 charging sessions and we can expect that the unsolved scenarios with 250 km EV range (the gap to 100% of instances) are also the longer ones. The decreasing planning time with fewer charging sessions is a good promise that with the future extension of EV ranges not only the usability of EVs themselves will

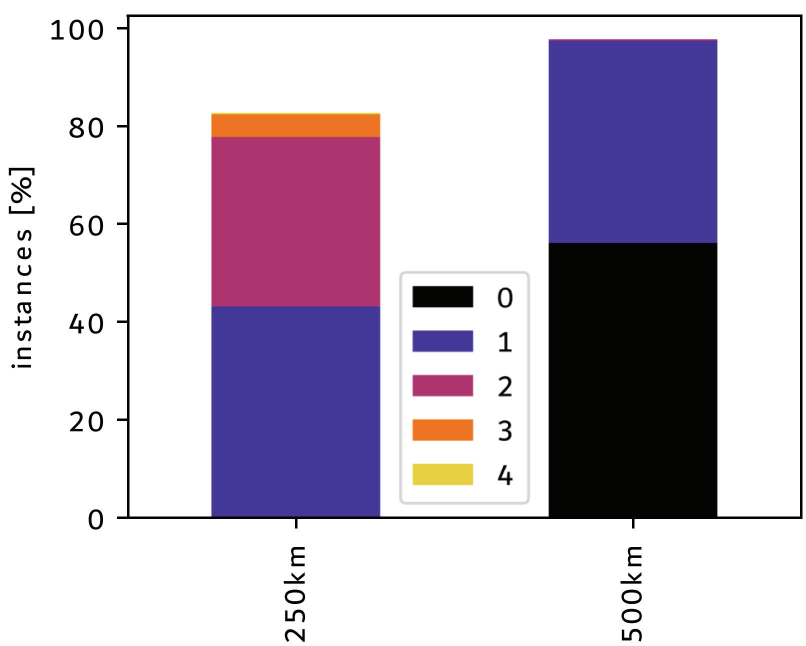


Figure 8. Distribution of the minimal number of charging sessions required to get from the request origin to the request destination for different EV ranges.

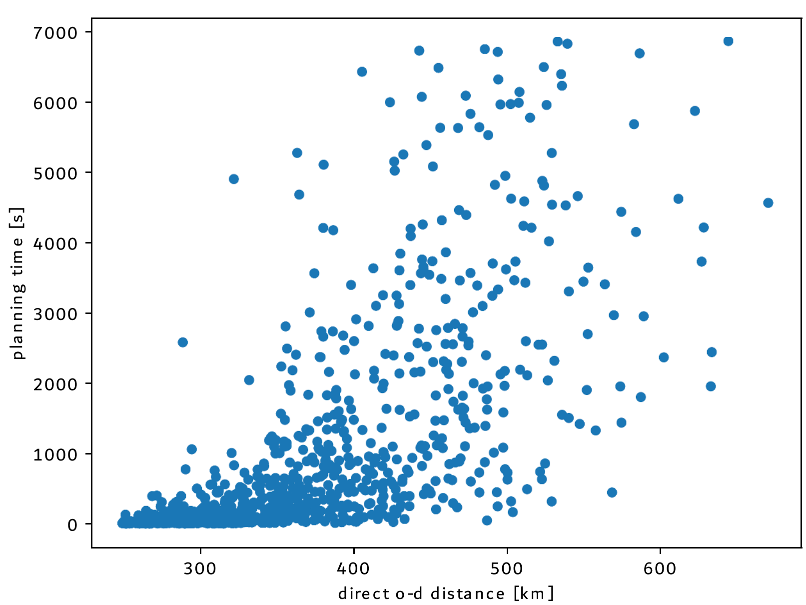


Figure 9. Dependency of the planning time on the distance between planning request origin and destination. Each dot represents one planning request from the evaluation request set.

improve but also the planning tools will be able to compute better plans on greater distances.

Figure 9 shows the dependency of planning time on the distance between origin and destination. We can see that with shorter distances the correlation is quite high but with increasing distance, the dependency weakens. It appears that around requests requiring at least 2 charging (with 250 km range approx. 500 km o-d distance) there is another influence, probably the availability of charging infrastructure.

### Evaluation results: approximate algorithm

To achieve faster planning times, we need to resort to an approximate algorithm based on a relaxed dominance relation. However, the $\varepsilon$-relaxation does not preserve optimality; therefore, we need to measure also the impact on the quality of produced travel plans. We ran the experiments on Germany planning environment with 24 combinations of $\varepsilon$ coefficients described in Evaluated algorithm configurations section.

The optimal algorithm is used as the baseline – which means that dimensionality reduction and time and cost heuristics are used. An overview of the algorithm and problem parameters can be seen in Table 4.

**Table 4.** Overview of used problem and algorithm parameters used for the evaluation of *informed approximate algorithm*.

| | |
|---|---|
| Problem Parameters | |
| Environment | **Germany**, Bavaria |
| CS pricing policy | **all** |
| EV range [km] | **250** |
| $\psi$ [€/km] | **0.03** |
| Algorithm Parameters | |
| Dimensionality reduction | **on** |
| Time heuristic $h_t$ | **on** |
| Cost heuristic $h_c$ | **on** |
| $\varepsilon_t$ | **1.0**, 0.995, 0.99, 0.98, 0.96, 0.93, 0.9 |
| $\varepsilon_c$ | **1.0**, 0.995, 0.99, 0.98, 0.96, 0.93, 0.9 |
| $\varepsilon_b$ | **1.0**, 0.995, 0.99, 0.98, 0.96, 0.93, 0.9 |

Note: Default parameters used unless stated otherwise are bold.

Figure 10 shows the trade-off between the achieved speed-up and the loss of solution quality. Quite unexpected is the essentially zero impact of the time component relaxation (the red cluster in the bottom left corner) on both planning time and solution quality. It is probably caused by the lesser impact of the relaxation due to the fact that the dimensionality reduction causes many dominance checks to ignore the time component.

A very good speed-up of approx. 50× can be achieved while maintaining a very reasonable solution quality loss of 0.05. In fact, a speed-up of more than 110× while getting 12 travel plans on average can be achieved with the quality loss ~0.08. To translate the speed-up to planning time, see Figure 11. For practically usable average planning times below 12s (median 6s) on the whole Germany area, we can use the $\varepsilon$ coefficients set to 0.9. The average Pareto-set size is 12 which provides a good variety of travel plans and the maximum planning time is 126 seconds which is very promising. On the simpler Bavaria planning environment, the average planning time is 4 seconds (median 2s, max 52s) even for the coefficients set to 0.995.

For a clearer image of how the relaxation impacts the solution quality, you can see Figure 12 displaying the optimal solution Pareto-set compared to the Pareto-set affected by the $\varepsilon$-relaxation (how these relaxed solutions look on a map can be seen in Figure 1). It dramatically reduces the number of found travel plans (from 41 to 7), but it still maintains a very good distribution. We can see that the individual plans in the relaxed Pareto-set are distributed among the clusters in the optimal Pareto-set as well as on the map.

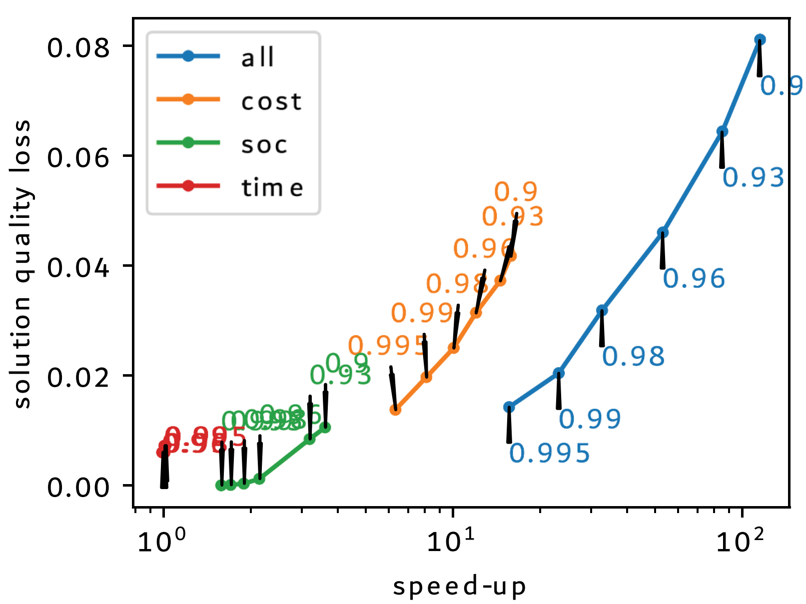


**Figure 10.** The trade-off between average solution quality loss and average speed-up achieved by $\varepsilon$-relaxation. The $\varepsilon$ coefficients are displayed directly in the plot. Lower quality loss and greater speed-up are better.

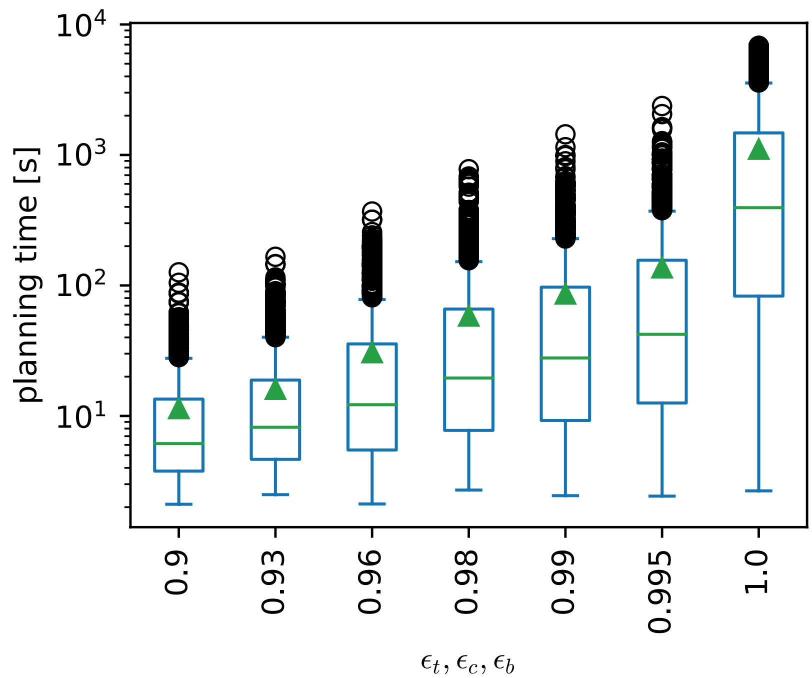


**Figure 11.** Planning time distribution of ε-relaxation with *all* coefficients set to the same value.

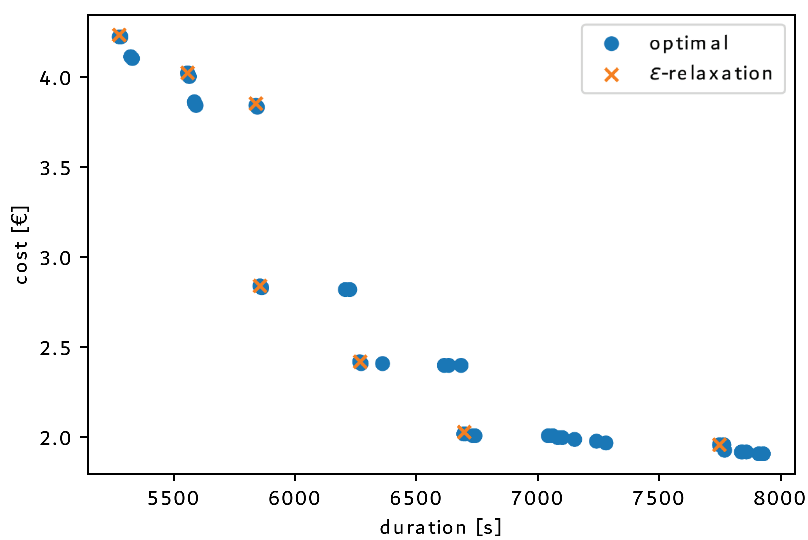


**Figure 12.** Comparison of the optimal pareto-set to the approximate solution pareto-set calculated by the algorithm for one request while using ε-relaxation with all ε coefficients set to 0.95.

Similarly to the informed optimal algorithm, we also evaluated the impact of problem parameters on the planning times of the approximate version of the algorithm. The results were very similar and we, therefore, do not present them in more detail here.

Although our algorithm is capable of solving the EV travel planning problem with time-dependent travel times and also with time-dependent charging pricing policies, we did not perform the evaluation due to the lack of relevant real-world data. However, we estimate that time-dependent travel times would have a negligible impact on the planning time since there would not be a significant change in the number of travel options (the number of roads is still the same). The time-dependent pricing policies would require introducing of waiting for cheaper charging price and therefore slightly increasing the number of charging options. However, we estimate the impact on planning time would still be insignificant.

## Conclusions

We introduced a novel multi-objective algorithmic approach to EV travel planning on realistic country-sized planning environments, which – in contrast to existing approaches – allows simultaneous optimization of multiple objectives and which generates Pareto-sets containing multiple and possibly many EV travel plans with different trade-offs between the values of the multiple travel planning objectives. Specifically, we explored the bi-objective case of simultaneously minimizing the time and the cost (in particular, the cost of charging) of EV travel plans.

Despite the very high computational complexity of the underlying travel planning problem, we managed to design an algorithm that achieves practically usable planning times with only a minor loss of solution quality. Our proposed algorithm uses several efficient speed-up techniques and can produce EV travel plans, on average, in less than 12 seconds on the whole Germany area and less than 4 seconds on the area of Bavaria. The algorithm achieved the planning times by the combination of informed search (heuristics), complexity reduction of the most time-consuming steps of the algorithms (dimensionality reduction), and the sub-optimal pruning of very similar solutions (epsilon-relaxation).

Moreover, we extensively evaluated the impact of key problem and algorithm parameters on the performance of our algorithms and on the quality of produced results, providing useful insights for future improvements. For example, as expected, the algorithms and EV users can benefit from the positive impact of a greater EV range with progress in battery technology. Unfortunately, the rising cost of electricity and consequently reduced options of free charging increases the complexity of the problem that is not mitigated enough by the cost heuristic as expected.

There are several directions in which our presented approach can be improved and/or extended. Road graph pre-processing techniques such as Contraction Hierarchies (Geisberger et al. 2012) can be used to further significantly accelerate the planning. Our approach can easily be adapted to accommodate other EV travel plan optimization objectives, such as the impact on EV battery health or the amount of renewable energy used for charging. Crucially, our approach can be straightforwardly extended into a time-dependent setting, where the travel time and/or the cost of charging vary dynamically with time. Even though the charging discretization has a very small impact on the resulting plans, research could be performed on whether it is possible to modify a technique proposed by Baum et al. (2019b) to multi-objective problems.

We believe our presented multi-objective approach paves the way toward powerful and flexible EV travel planning systems that would make electromobility more efficient and accessible to the widest range of users.

## Notes

1. As illustrated by the EV road trip of US secretary of energy in 2023 https://www.npr.org/2023/09/10/1187224861/electric-vehicles-evs-cars-chargers-charging-energy-secretary-jennifer-granholm or repeatedly voiced concerns of EV drivers interviewed by Business Insider https://www.businessinsider.com/electric-vehicle-owners-advice-realities-cost-maintenance-driving-electric-2023-11?op=1\#charlotte-scot-76-old-lyme-ct-kia-ev-6-1
2. Such as the built-in navigation tool in the Tesla EVs.
3. We use the term *EV travel planning* as an umbrella term for route planning combined with planning of charging for EV drivers, as described in this introduction.
4. http://its.fel.cvut.cz/ev-travel-planner. Note that the application does not use precise travel time data (they are expensive) and is for the purpose of potential capabilities and usage demonstration only.
5. *Route planning* (Bast et al. 2016) can be seen as a specific case of *route optimization* where the focus is on finding the optimal point-to-point route/plan for only one vehicle or passenger.
6. Although the reconstruction of the final plans requires additional state attributes (for example, a reference to the preceding state and charging details), we omitted them for a clearer presentation.
7. The algorithm maintains open and closed sets for all nodes to contain only non-dominated states.
8. In theory, the optimal solution of the EV travel planning problem would require the ability to consider any arbitrary target charging level. In practice, however, the user can only choose from a discrete set of target charging levels when charging the vehicle and the discretization of the target charging level can be considered as part of the definition of the EV travel planning problem. For this reason, and to simplify the exposition, we refer to EV travel planning as optimal as long as it is optimal with regards also to the set of predefined charging levels.
9. https://download.geofabrik.de/europe/germany.html
10. https://chargemap.com
11. https://www.earthdata.nasa.gov/sensors/srtm
12. For each free charging station, we randomly selected a paid one with the same power and used its pricing policy.
13. All speed-ups are comparisons of the full optimal algorithm without the one specific component. Because of the insufficient performance and too small number of completed instances, we cannot reliably calculate the speed-up on the Germany planning environment.
14. The boxplots show median (green line), mean (green triangle), the box showing Q1 (the 25th percentile) and Q3 (the 75th percentile), and the whiskers show the lowest and highest point within 1.5 IQR of the lower and higher quartile respectively. The outliers are shown as circles.
15. We call a newly generated state a *successor* of the original state. It has the transitive property – meaning that a successor of a successor is also a successor of the first state.
16. First-in first-out: a worse input value cannot generate a better result. For example, a later departure cannot lead to earlier arrival.
17. The battery capacity maximum in SoC calculation does not invalidate the FIFO property.
18. Charging of greater amount of energy resulting on the same SoC has to be slower.
19. The only possibly negative attribute is the consumption/SoC, and there cannot be a negative consumption cycle due to the law of physics.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Funding

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS22/168/OHK3/3T/13.

## ORCID

Marek Cuchý http://orcid.org/0000-0002-4930-7500

## References

Aneja, Y. P., and K. P. K. Nair. 1978. “The Constrained Shortest Path Problem.” *Naval Research Logistics Quarterly* 25 (3): 549–555. http://doi.org/10.1002/nav.3800250314.

Artmeier, A., J. Haselmayr, M. Leucker, and M. Sachenbacher, 2010. “The Shortest Path Problem Revisited: Optimal Routing for Electric Vehicles.” In *Annual Conference on Artificial Intelligence*, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, 309–316. Springer.

Bast, H., D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. F. Werneck. 2016. “Route Planning in Transportation Networks.” In *Algorithm Engineering*, edited by L. Kliemann and P. Sanders, 19–80. Cham: Springer.

Batista, L. S., F. Campelo, F. G. Guimarães, and J. A. Ramírez, 2011. “A Comparison of Dominance Criteria in Many-Objective Optimization Problems.” In *Evolutionary Computation (CEC), 2011 IEEE Congress on Evolutionary Computation*, New Orleans, LA, USA, 2359–2366. IEEE.

Bauer, R., and D. Delling. 2009. “Sharc: Fast and Robust Unidirectional Routing.” *Journal of Experimental Algorithmics (JEA)* 14:2–4. https://doi.org/10.1145/1498698.1537599.

Baum, M., J. Dibbelt, A. Gemsa, D. Wagner, and T. Zündorf. 2019. “Shortest Feasible Paths with Charging Stops for Battery Electric Vehicles.” *Transportation Science* 53 (6): 1627–1655. https://doi.org/10.1287/trsc.2018.0889.

Baum, M., J. Dibbelt, T. Pajor, J. Sauer, D. Wagner, and T. Zündorf. 2019. “Energy-Optimal Routes for Battery Electric Vehicles.” *Algorithmica* 82 (5): 1490–1546. https://doi.org/10.1007/s00453-019-00655-9.

Bellman, R. 1958. "On a Routing Problem." *Quarterly of Applied Mathematics* 16 (1): 87–90. https://doi.org/10.1090/qam/102435.
Ben Abbes, S., L. Rejeb, and L. Baati. 2022. "Route Planning for Electric Vehicles." *IET Intelligent Transport Systems* 16:875–889.
Delling, D., and D. Wagner. 2009. "Pareto Paths with SHARC." In *8th International Symposium on Experimental Algorithms*, Tech Univ Dortmund, Dortmund, Germany, 125–136. Springer.
Dijkstra, E. W. 1959. "A Note on Two Problems in Connexion with Graphs." *Numerische mathematik* 1 (1): 269–271. https://doi.org/10.1007/BF01386390.
Eisner, J., S. Funke, and S. Storandt, 2011. "Optimal Route Planning for Electric Vehicles in Large Networks." In *Proceedings of the 25th Conference on Artificial Intelligence*, San Francisco, CA, USA, 1108–1113.
Eppstein, D., and M. T. Goodrich, 2008. "Studying (Non-Planar) Road Networks Through an Algorithmic Lens." In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, Irvine, CA, USA, 1–10.
Garey, M. R., and D. S. Johnson. 1979. *Computers and Intractability*. Vol. 174. San Francisco, CA, USA: freeman San Francisco.
Geisberger, R., P. Sanders, D. Schultes, and C. Vetter. 2012. "Exact Routing in Large Road Networks Using Contraction Hierarchies." *Transportation Science* 46 (3): 388–404. https://doi.org/10.1287/trsc.1110.0401.
Hart, P. E., N. J. Nilsson, and B. Raphael. 1968. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." *IEEE Transactions on Systems Science and Cybernetics* 4 (2): 100–107. https://doi.org/10.1109/TSSC.1968.300136.
Hrnčíř, J., P. Žilecký, Q. Song, and M. Jakob. 2016. "Practical Multicriteria Urban Bicycle Routing." *IEEE Transactions on Intelligent Transportation Systems* 18 (3): 493–504. https://doi.org/10.1109/TITS.2016.2577047.
Johnson, D. B. 1977. "Efficient Algorithms for Shortest Paths in Sparse Networks." *Journal of the ACM (JACM)* 24 (1): 1–13. https://doi.org/10.1145/321992.321993.
Laszczyk, M., and P. B. Myszkowski. 2019. "Survey of Quality Measures for Multi-Objective Optimization: Construction of Complementary Set of Multi-Objective Quality Measures." *Swarm and Evolutionary Computation* 48:109–133. https://doi.org/10.1016/j.swevo.2019.04.001.
Liu, Y., X. Feng, L. Zhang, W. Hua, and K. Li. 2020. "A Pareto Artificial Fish Swarm Algorithm for Solving a Multi-Objective Electric Transit Network Design Problem." *Transportmetrica A: Transport Science* 16 (3): 1648–1670. https://doi.org/10.1080/23249935.2020.1773574.
Mandow, L., J. P. De la Cruz. 2005. "A New Approach to Multiobjective A* Search." In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland.
Martins, E. Q. V. 1984. "On a multicriteria shortest path problem." *European Journal of Operational Research* 16 (2): 236–245. https://doi.org/10.1016/0377-2217(84)90077-8.
Möhring, R. H., H. Schilling, B. Schütz, D. Wagner, and T. Willhalm. 2007. "Partitioning graphs to speedup dijkstra's algorithm." *Journal of Experimental Algorithmics (JEA)* 11:2–8. https://doi.org/10.1145/1187436.1216585.
Müller-Hannemann, M., and K. Weihe. 2006. "On the Cardinality of the Pareto Set in Bicriteria Shortest Path Problems." *Annals of Operations Research* 147 (1): 269–286. https://doi.org/10.1007/s10479-006-0072-1.
Pulido, F. J., L. Mandow, and J. L. Pérez-de-la Cruz. 2015. "Dimensionality Reduction in Multi- Objective Shortest Path Search." *Computers & Operations Research* 64:60–70. https://doi.org/10.1016/j.cor.2015.05.007.
Rajan, P., M. Baum, M. Wegner, T. Zündorf, C. J. West, D. Schieferdecker, and D. Delling, 2021. "Robustness Generalizations of the Shortest Feasible Path Problem for Electric Vehicles." In *21st Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2021)*, Online, Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
Riquelme, N., C. Von Lücken, and B. Baran, 2015. "Performance Metrics in Multi-Objective Optimization." In *2015 Latin American computing conference (CLEI)*, SPC, UNSA, UCSM, UCSP, ULS, Arequipa, Peru, 1–11. IEEE.
Sachenbacher, M., M. Leucker, A. Artmeier, and J. Haselmayr, 2011. "Efficient Energy-Optimal Routing for Electric Vehicles." In *Proceedings of the 25th Conference on Artificial Intelligence*, San Francisco, CA, USA.
Sanders, P., and D. Schultes. 2006. *Engineering Highway Hierarchies*, 804–816. in: ESA, Springer.
Schoenberg, S., and F. Dressler. 2023. "Reducing Waiting Times at Charging Stations with Adaptive Electric Vehicle Route Planning." *IEEE Transactions on Intelligent Vehicles* 8 (1): 95–107. https://doi.org/10.1109/TIV.2022.3140894.
Schönfelder, R., M. Leucker, and S. Walther, 2014. "Efficient Profile Routing for Electric Vehicles." In *International Conference on Internet of Vehicles*, Beijing, China, 21–30. Springer.
Siddiqi, U. F., Y. Shiraishi, and S. M. Sait, 2011. "Multi-Constrained Route Optimization for Electric Vehicles (Evs) Using Particle Swarm Optimization (Pso)." In *2011 11th International Conference on Intelligent Systems Design and Applications*, Cordoba, Spain, 391–396. IEEE.
Stewart, B. S., and C. C. White III. 1991. "Multiobjective a." *Journal of the ACM (JACM)* 38 (4): 775–814. https://doi.org/10.1145/115234.115368.
Storandt, S., 2012. "Quick and Energy-Efficient Routes: Computing Constrained Shortest Paths for Electric Vehicles." In *Proceedings of the 5th ACM SIGSPATIAL international workshop on computational transportation science*, Redondo Beach, CA, USA, 20–25.
Storandt, S., and S. Funke, 2012. "Cruising with a Battery-Powered Vehicle and Not Getting Stranded." In: Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, Ontario, Canada.
Tran, C. Q., M. Keyvan-Ekbatani, D. Ngoduy, and D. Watling. 2021. "Stochasticity and Environ- Mental Cost Inclusion for Electric Vehicles Fast-Charging Facility Deployment." *Transportation Research Part E: Logistics & Transportation Review* 154:102460. https://doi.org/10.1016/j.tre.2021.102460.
Ünal, V., M. Soysal, M. Çimen, and Ç. Koç. 2022. "Dynamic Routing Optimization with Electric Vehicles Under Stochastic Battery Depletion." *Transportation Letters* 15 (10): 1376–1388. https://doi.org/10.1080/19427867.2022.2157365.
Zhu, S. 2022. "Multi-Objective Route Planning Problem for Cycle-Tourists." *Transportation Letters* 14 (3): 298–306. https://doi.org/10.1080/19427867.2020.1860355.
Zündorf, T., 2014. "Electric vehicle routing with realistic recharging models." Master's thesis. Department of Informatics Karlsruhe, Germany.

# Appendices

## Appendix A. Optimal algorithm properties

To prove that the optimal algorithm with both heuristics and dimensionality reduction finds the whole Pareto-optimal set of travel plans, we first need to prove that a simplified *uninformed* version of the optimal algorithm without any heuristic and dimensionality reduction (which is basically a multi-objective extension of Dijkstra's algorithm) finds the whole Pareto-optimal set of travel plans. We provide the proof in Appendix A.1. Since the *uninformed* algorithm finds the whole Pareto-optimal set of travel plans and the heuristics used to extend it to an A*-based algorithm are consistent (see proofs in Appendix B), the algorithm improved with the heuristics also finds the whole Pareto-optimal set of travel plans. The requirements of the dimensionality reduction posed by Pulido et al. (2015) are also satisfied – lexicographical ordering of the priority queue and heuristic consistency. Therefore, the complete *optimal* algorithm as proposed in Optimal EV travel planning algorithm section also finds the whole Pareto-optimal set of travel plans.

## Appendix A.1. *Uniformed algorithm properties*

In this section, we examine and prove the properties of the *uninformed algorithm*, which is a simplified version of the optimal algorithm described in Optimal EV travel planning algorithm section without heuristics and dimensionality reduction. At first, we prove that all the plans/states it finds are non-dominated by each other. Then, we prove that it finds all plans that are Pareto-optimal with regards to the problem definition and the restriction posed on the charging actions by the predefined target charging levels. Finally, we prove that the algorithm terminates in a finite number of steps.

**Theorem 1**. The base uninformed algorithm has the label/state setting property.
The label/state setting property means that when a state is extracted (settled) from the priority queue, it will not be dominated by any other state at the same node that is extracted from the queue later. That means that the state is Pareto-optimal with regards to all other states at the same node.

**Proof.** Assume that the extracted state $s = \langle v, t, c, b \rangle$ will be dominated. It means that a state $s' = \langle v', t', c', b' \rangle$ at the same node ($v' = v$) with all the following properties has to appear in the queue: lower or equal time ($t' \leq t$), lower or equal cost ($c' \leq c$), and higher or equal SoC ($b' \geq b$). State at the same node with all attributes equal cannot appear in the queue since it would be dominated by $s$ in the closed set $S_v^{\mathrm{cl}}$ and pruned (line 22 in Algorithm 1). Therefore, state $s'$ has to be strictly better at at least one of the attributes. Since state $s$ is lexicographically smallest in the queue, state $s'$ cannot be in the queue and is yet to be generated. It means that state $s'$ has to be generated from a state $s'' = \langle v'', t'', c'', b'' \rangle$ that is already in the queue, or from the extracted state $s$ itself (creating a cycle), or from any of their successors.[15] We will prove that any state generated from $s''$, $s$, or any of their successors cannot dominate $s$. State $s''$ is lexicographically greater or equal to the lexicographically smallest label $s$. Therefore, it has:

(1) higher time ($t'' > t$), or
(2) equal time ($t'' = t$) and higher cost ($c'' > c$), or
(3) equal time ($t'' = t$), cost ($c'' = c$), and smaller or equal SoC ($b'' \leq b$).

The first two cases cannot generate a state that dominates state $s$, since both time and cost objectives are non-decreasing. In the third case, it is possible to improve the state of charge (by recuperation or charging), but it is impossible without increasing the time objective since it is impossible to move or charge in no time. The third case also applies to the expansion of the extracted state itself (everything is equal), therefore, also in this particular case a dominating state cannot be generated. Following all of the above, the successors of $s''$, or $s$ cannot be lexicographically smaller than $s$. Therefore, also the successors cannot generate a state dominating $s$.
Therefore, it is impossible to add a state dominating $s$ to the queue.

Since any state extracted from the priority queue is Pareto-optimal with regards to the states at the same node, the extracted states at the destination node are also Pareto-optimal.

**Lemma 1.** Let a sub-plan $\pi_i^*$ of a Pareto-optimal plan $\pi^* = (s_0, a_0, s_1, a_1, \ldots, a_{k-1}, s_k)$ be its sub-sequence of states and actions $(s_0, a_0, s_1, a_1, \ldots, a_{i-1}, s_i)$, where $i < k$. Each sub-plan $\pi_i^*$ of a Pareto-optimal plan $\pi^*$ is also Pareto-optimal with regards to all (sub-)plans ending at node $v_i$.

**Proof.** We will prove the Pareto-optimality of sub-plans by proving that it is impossible to generate a Pareto-optimal (sub-)plan from a dominated (sub-optimal) sub-plan, therefore, it is also impossible to generate a Pareto-optimal solution plan $\pi^*$ from a dominated sub-plan.

Assume that $s = \langle v, t, c, b \rangle$ is the end state representing the dominated sub-plan, which means that there exists a state (representing another sub-plan) $s' = \langle v', t', c', b' \rangle$ such that $s' \preceq s \land v = v'$. We claim that there does not exist an action $a$ applied to $s$ that would generate a state $s_a = \langle v_a, t_a, c_a, b_a \rangle$ that would not be dominated by a state $s'_a = \langle v'_a, t'_a, c'_a, b'_a \rangle$ generated by the same action applied to $s'$ or already by the state $s'$ itself. We prove it for *move* actions and separately for *charge* actions.
The above holds for all *move* actions $a$ because they satisfy FIFO[16] property since they are defined solely by additions and subtraction of constant values defined by the road graph edges.[17] Therefore, it is impossible to achieve non-dominated state attributes with dominated initial state by a move action ($s' \preceq s \Rightarrow s'_a \preceq s_a$).
The *charging* actions are generated for the predefined set of target charging levels. For the purpose of the proof, we define the set of target charging levels as $L \subset (0, b_{\max}]$. We can divide the charging actions in two cases by the considered target charging level $l \in L$:

*Case 1*: $b < l \leq b'$. Since the charging function $\phi$ is positive and $t \geq t'$ (from the definition of dominance), then $t_a > t'$. Since all charging cost functions $\gamma_q$ are non-negative and $c \geq c'$, then $c_a \geq c'_a$. And since $b_a = l \leq b'$, then $s' \preceq s_a$.

*Case 2*: $l > b'$. We define the amount of charged energy for the dominated state $s$ as $j = l - b$ and for the dominating state $s'$ as $j' = l - b'$ while $l = b_a = b'_a$. Since $b \leq b'$, then $j \geq j'$. When a target charging level (and charging power $P$) is equal, the charging function $\phi$ satisfies the FIFO property.[18] Therefore, $\phi(b, l, P) \geq \phi(b', l, P)$. And since, $t \geq t'$ and $t_a = t + \phi(b, l, P)$ (accordingly for $t'_a$), then $t_a \geq t'_a$. Since both charged energy and charging duration are greater for the dominated state $s$, none of the charging cost functions $\gamma_q$ can yield a cheaper charging ( $q \in Q : \gamma_q(\phi(b, l, P), j) \geq \gamma_q(\phi(b', l, P), j')$; and since $c \geq c'$, then $c_a \geq c'_a$. And because $b_a \geq b'_a$, then $s'_a \preceq s_a$ .

**Definition 6.** Let $\Pi^*$ be the non-dominated set of all plans solving the given problem instance: there does not exist a plan $\pi \notin \Pi^*$ ending in the destination such that $0 = \exists 0 \not\pi^* \in \Pi^* : \pi^* \pi$.

**Theorem 2.** The base uninformed algorithm finds $\Pi^*$.

**Proof.** Since all sub-plans of a Pareto-optimal solution plan are Pareto-optimal (Lemma 1), they cannot be pruned by any of the dominance checks during the search and since the algorithm explores all possible actions for each extracted state, we can prove that the plan is found. It first expands the state $s_0$ by action $a_0$ (among others) leading to state $s_1$. Since state $s_1$ is Pareto-optimal, it cannot be pruned from the priority queue; and therefore, will be expanded eventually. The expansion by action $a_1$ leads to state $s_2$ that again cannot be pruned from the priority queue. This procedure is repeated until $s_k$ is extracted from the queue.

**Theorem 3.** The base uninformed algorithm ends in a finite number of steps.

**Proof.** Since the road graph is finite and there are no negative cycles,[19] none of the dominance attributes (time, cost, SoC) can improve infinitely; therefore, eventually all states generated by *expand* function will reach the destination or be pruned by the closed/settled states in corresponding Pareto-set $S_v^{cl}$ and/or solution states $\Pi$.

## Appendix B. Heuristics details and consistency

**Definition 7.** Let $s, s'$ be two consecutive states in a *planning problem search space*. We say that heuristic $h$ for a single objective is *consistent* iff $h(s) \leq c(s, s') + h(s')$, where $c(s, s')$ is the objective cost for the transition between the two states.

### Appendix B.1. *Remaining travel time heuristic*

**Theorem 4.** The heuristic $h_t$ is consistent.

**Proof.** To prove the consistency of the heuristic, we need to prove the following condition is true for all consecutive states $s, s'$:

$$t(v, v_{\text{goal}}) \leq g(s, s') + t(v', v_{\text{goal}}) \tag{B.1}$$

where $g$ is the transition time cost between the two states. Note that the cost is non-negative.
In the case of the *move* action, the condition is always true since it is essentially the triangle inequality, which the shortest path satisfies.

Now, we prove the consistency condition also holds for *charge* action. We prove this by contradiction. If any charge action violated the condition, there would have to exist a charge action (defined by the two states $s, s'$) such that: $t(v, v_{\text{goal}}) > g(s, s') + t(v', v_{\text{goal}})$. During a charge action the location does not change ($v = v'$), therefore the heuristic also does not change: $t(v, v_{\text{goal}}) > g(s, s') + t(v, v_{\text{goal}}) \Leftrightarrow 0 > g(s, s')$. Since the transition cost cannot be negative, we have a contradiction. Therefore, all *charge* actions satistify the consistency condition.

There are no transitions other than *move* and *charge* actions. Therefore, all transitions satisfy the consistency condition.

### Appendix B.2. *Minimum remaining charging and driving cost heuristic*

The minimum price per amount of energy $c_{\min}$ calculation cannot be easily described for the general case since both the charging function and the cost function consider too many parameters. However, it is possible to do so for a specific case. For example, suppose the cost function is duration-based (for instance, per minute of charging), and the charging function is piecewise linear. In that case, we can find the segment of the charging function where the charging function is the most efficient and compute the minimum price per unit of energy based solely on the most efficient segment of the charging function. Calculation of the minimum price is also simple for the fixed price per charging session, i.e. the same cost no matter how much energy is charged or how long the charging takes. In such a case, the minimum price per amount of energy is achieved while charging the battery from complete depletion to the maximum capacity. Therefore, we divide the fixed price by the battery capacity.

**Theorem 5.** The heuristic $h_c$ is consistent.

**Proof.** To prove the consistency of the heuristic, we need to prove the following condition is true for all consecutive states $s, s'$:

$$(\beta(v, v_{\text{goal}}) - b)c_{\min} + \psi d(v, v_{\text{goal}}) \leq g(s, s') + (\beta(v', v_{\text{goal}}) - b')c_{\min} + \psi d(v', v_{\text{goal}}) \tag{B.2}$$

where $g$ is the transition money cost between the two states. Note that the cost $g$ and the minimum price $c_{\min}$ are non-negative.
We again split the proof based on the two possible actions: *move* and *charge*. To prove the consistency for all *move* actions, we split the original inequality condition B.2 into two separate inequalities (one for charging cost part and one for driving cost part) that if both are satisfied, also the original inequality is satisfied:

$$\psi d(v, v_{\text{goal}}) \leq g(s, s') + \psi d(v', v_{\text{goal}}) \tag{B.3}$$

$$\wedge$$

$$(\beta(v, v_{\text{goal}}) - b)c_{\min} \leq (\beta(v', v_{\text{goal}}) - b')c_{\min} \tag{B.4}$$

$$\Rightarrow$$

$$(\beta(v, v_{\text{goal}}) - b)c_{\min} + \psi d(v, v_{\text{goal}}) \leq g(s, s') + (\beta(v', v_{\text{goal}}) - b')c_{\min} + \psi d(v', v_{\text{goal}})$$

Since all *move* actions cost $g(s, s') = \psi d(v, v')$, the inequality B.3 is equivalent to: $\psi d(v, v_{\text{goal}}) \leq \psi d(v, v') + \psi d(v', v_{\text{goal}})$. If the cost per km $\psi = 0$, the inequality is trivially true. If $\psi > 0$, the inequality simplifies to $d(v, v_{\text{goal}}) \leq d(v, v') + d(v', v_{\text{goal}})$, which is the triangle inequality for shortest path and therefore satisfied. In the case the minimum price $c_{\min}$ in inequality B.4 is 0, the condition is trivially true. In the case the minimum price is positive, the inequality simplifies to: $\beta(v, v_{\text{goal}}) - b \leq \beta(v', v_{\text{goal}}) - b' \Leftrightarrow \beta(v, v_{\text{goal}}) \leq \beta(v', v_{\text{goal}}) + b - b'$. Since the SoC difference $b - b'$ can be seen as the energy transition cost between the two states (and one edge in the graph), the last inequality is essentially the triangle inequality with the most energy-efficient paths. Even though the energy cost of an edge can be negative, there are no negative cycles; therefore, the inequality is still satisfied, and therefore, the inequality B.4 is true for all *move* actions. Since both inequalities (B.3 and B.4) are true for all *move* actions, also the original inequality condition B.2 is true for all *move* actions.

During a *charge* action, the location does not change ($v = v'$), therefore: $(\beta(v, v_{\text{goal}}) - b)c_{\min} + \psi d(v, v_{\text{goal}}) \leq g(s, s') + (\beta(v, v_{\text{goal}}) - b')c_{\min} + \psi d(v, v_{\text{goal}}) \Leftrightarrow \beta(v, v_{\text{goal}})c_{\min} - bc_{\min} \leq g(s, s') + \beta(v, v_{\text{goal}})c_{\min} - b'c_{\min} \Leftrightarrow (b' - b)c_{\min} \leq g(s, s')$. Since $c_{\min}$ is the minimum achievable price per energy among all the charging stations and $b' - b$ is the amount of charged energy during the charge action, the cost of the charge action $g(s, s')$ cannot be smaller, and the inequality holds.

## Appendix C. Implementation notes

We implemented all variants of our EV travel planning algorithm in Java 17. We did not focus solely on the performance during the development but also considered the maintainability of the code base. We made this decision for the sake of easy configuration and easy replacement of various algorithm components because we experimented with numerous variants of the algorithm. We estimate the performance penalty of such a modular implementation to be between 10% and 20%.

One of the most frequent operations is the dominance checks by a Pareto-set (line 22 of Algorithm 1). A naive method is to iterate over the whole Pareto-set. We used a better approach that keeps the Pareto-sets lexicographically ordered, allowing the multi-level binary search to skip part of the Pareto-set. It proves to be faster even though it is computationally non-trivial to maintain the ordering since insertion/deletion into/from the middle requires copying of the trailing part of the array. The ordering is especially useful with Pareto-sets based on only two state attributes. It allows to skip an even bigger part of the Pareto-set due to the property that all/both relevant state attributes are in ascending/descending order (unlike with three attributes where we do not know anything about the ordering of the third attribute). Two state attributes are used, for example, in the dominance checks with closed Pareto-set $S_v^{\mathrm{cl}}$ (line 22) simplified by the dimensionality reduction.

Unfortunately, the simple union of opened Pareto-sets $S^{\mathrm{op}}$ is not suitable for finding the minimum state (line 11 in Algorithm 1). Therefore, we use a separate data structure for this purpose – a binary heap. We update the opened Pareto-sets $S_v^{\mathrm{op}}$ and the binary heap simultaneously. Whenever a state is added/removed to/from one structure, it is also added/removed to/from the other.

We also efficiently pre-calculate all the essential information required by both heuristics at once before the core algorithm starts. All travel times required by the time heuristic are computed by running backward single-objective label-setting Dijkstra's algorithm (Dijkstra 1959) from the destination node $v_{\mathrm{goal}}$ to all other nodes optimizing travel time, which is very fast (compared to the core of the algorithm). We use the same algorithm to calculate the distance of the shortest path required by the cost heuristic. However, to calculate the minimum required energy, we run a label-correcting version of Dijkstra's algorithm from the destination node $v_{\mathrm{goal}}$ to all other nodes optimizing energy consumption. The label-correcting version of the algorithm is required since the energy consumption on an edge can be negative. The label-correcting version of Dijkstra's algorithm is slower than the common label-setting version, but it is still very fast compared to the core algorithm.

## Appendix D. Evaluation EV travel problem instances details

### Appendix D.1. *Evaluation EV planning environments*

**Table A1.** The sizes of the EV travel planning environments used in the evaluation: the number of nodes ($|V|$) and the number of edges ($|E|$) of the road network graphs, and the number of charging stations ($|Q|$).

| environment | $\lvert V\rvert$ | $\lvert E\rvert$ | $\lvert Q\rvert$ |
|---|---|---|---|
| Germany | 1.5M | 3M | 12633 |
| Bavaria | 300k | 600k | 2225 |

The EV travel planning environments summary can be seen in Table A1.

The charging station dataset contains four different types of pricing policies:

- **energy** – The cost is based on the amount of energy the charging station provides, for example, per kilowatt-hour.
- **duration** – The cost is based on charging duration, for example, per minute.
- **fixed** – The user pays a fixed cost no matter how long it takes or how much energy is charged.
- **occupancy** – This policy is very similar to the duration-based policy. The difference is that there is a free period of time during which the occupancy fee is not paid, for example, the first two hours. The reason behind this fee is to motivate users to move out as soon as possible after the EV is charged and to not block the charging station.

The pricing policies can also be combined. For example, the charging cost can depend on the charging duration and also on the amount of energy charged. The resulting charging cost is a sum of the cost produced by both policies. Almost all (99%) of paid charging stations employ the *energy* based policy, the *duration* based policy is used on 27% of paid CS, the *fixed* price on 18%, and the *occupancy* based policy on 26% of paid CS.

Since we assume the users will not be willing to wait too long to charge their EVs, we have filtered out charging stations with a maximum charging power of less than 11 kW.

### Appendix D.2. *Evaluation EV model*

We represent the *energy consumption* with a linear model that takes into account the length and the elevation profile of the roads similarly to Eisner et al. (2011): $\beta(e) = \kappa d(e) + \lambda elev^{+}(e) - \alpha elev^{-}(e)$ where $elev^{+}$ and $elev^{-}$ are the elevation gain and elevation drop accumulated during the traversal of the edge $e$ in meters. We set the coefficients to $\kappa = 0.16, \lambda = 1.6$ and $\alpha = 1.2$ that with 40 kWh *battery capacity* lead to approx. 250 km range and with 80 kWh battery to 500 km range. The model is rather simple; however, the algorithms are consumption model agnostic and can be easily adapted for more complex realistic models with negligible impact on the algorithm performance compared especially to the range influence.

The algorithm we proposed works with arbitrary *charging functions*. For the evaluation, we chose a piecewise linear function similar to Baum et al. (2019b).

To describe the specific charging function, we need to describe *charging efficiency*. It can be written as:

$$\frac{charged\ energy}{power \cdot time}$$

For example, if we want to charge 10 kWh on a charging station with 10 kW power, the charging would take one hour with 100% efficiency. If the efficiency was 50%, it would take 2 hours.

We model the charging up to 80% of the battery capacity to be 99% efficient, from 80% to 85% to be 86% efficient, from 85% to 90% to be 63% efficient, from 90% to 95% to be 43% efficient and above 95% to be only 15% efficient (see Figure 3). It means that it takes more than 6.6× more time to charge the last 5% of the battery than it takes to charge the same amount of energy while the state of charge is below 80%. The parameters are based on the data presented in Zündorf (2014). Such a non-linear model significantly improves the accuracy of charging time estimates, which is essential for accurate EV travel planning.

### Appendix D.3. *Evaluation EV travel planning requests*

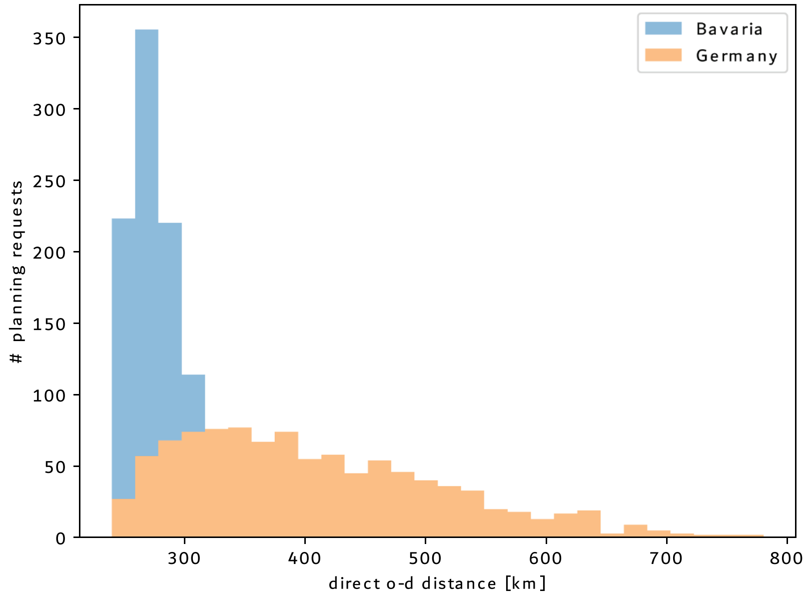


**Figure A1.** The distribution of the direct origin-destination distance for generated travel planning requests.