Enabling E-Mobility: One Way, Return, and with Loading Stations

Sabine Storandt
FMI, Universität Stuttgart
Stuttgart, Germany

Jochen Eisner
FMI, Universität Stuttgart
Stuttgart, Germany

Stefan Funke
FMI, Universität Stuttgart
Stuttgart, Germany

1. INTRODUCTION

Electric Vehicles (EVs) exhibit some important advantages compared to classical, fuel driven cars: Their 'fuel' can be produced from regenerative sources like wind, hydropower, or solar energy. In use, there is the possibility to recuperate energy during deceleration phases or when going downhill. Furthermore EVs typically exhibit lower emissions to their immediate environment in terms of combustion gases or noise levels.

On the other hand, current EV technology still suffers from some inconveniences which prevent an even faster acceptance of E-mobility. Due to weight and space constraints, EVs only have a limited energy reservoir constraining their cruising ranges, recharging a battery equipped EV typically takes quite a long time, and due to the non-ubiquity of charging stations is not always possible.

From a Computer Science point of view, these difficulties pose interesting questions where algorithmic methods and techniques can play a key role in enabling E-Mobility. In this short abstract we will highlight important algorithmic tools and how they solve fundamental problems in E-Mobility.

2. ENERGY-EFFICIENT ROUTE PLANNING

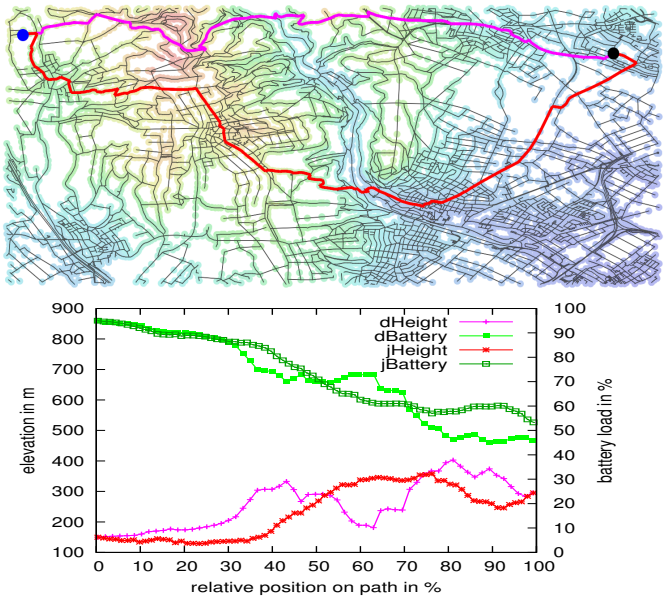

Figure 1: Shortest and most energy efficient path; battery load and elevation profiles.

The most basic problem to be solved in the context of E-Mobility is the minimization of energy consumption when planning a route from s to t. While for ordinary vehicles this is more a desired goal than a necessity, for EVs their limited energy reservoir combined with long recharging times make energy-aware route planning indispensable – even reaching a target at all might fail if energy runs out on the way. Algorithmically, route planning problems are always formulated in terms of a graph $G(V, E)$ with associated edge costs $cost : E \to \mathbb{R}$, in our case representing the energy consumption (if > 0) or recuperation (if < 0) along a road segment; G is free of negative cycles for obvious reasons. For a maximum battery capacity of $M \in \mathbb{R}^+$, an initial battery charge status $I \in [0, M]$, and source and target vertices $s, t \in V$ we are interested in finding a 'good' path from s to t. We will discuss several reasonable notions of 'good path' and outline efficient ways to answer respective queries.

2.1 Minimizing Net Energy Consumption

The most natural notion of 'good path' is one which minimizes the net energy consumption, i.e., for a given initial charge status $I \in [0, M]$, we want to find a path in G from s to t resulting in the largest possible battery level in t. While for given edge weights this appears like an ordinary shortest path problem in graphs, there are two complications caused by the nature of EVs. First, at no point along the path, the EV must run out of energy. Second, when – while driving along the path – the battery gets fully charged due to energy recuperation, the battery level will never exceed the maximum battery capacity M. These *battery constraints* together with the partially negative edge costs prohibit the use of conventional route planning algorithms. Artmeier et al. introduced this problem in [1] and presented an extension of the Bellman-Ford algorithm to solve the problem in time $\mathcal{O}(nm)$ with $|V| = n, |E| = m$ prohibiting its applicability for larger problem instances or on computationally weak (e.g. mobile) devices. In [3] we presented techniques that improved both theoretical as well as practical running times by orders of magnitude.

Our strategy was to first model the battery constraints as edge cost functions (depending on the battery charge status) which satisfy the FIFO property, hence allowing straightforward use of the Bellman-Ford algorithm. Then we shifted the partly negative edge costs using an adaptation of the potential function technique of Johnson [2] allowing the application of Dijkstra's algorithm without changing the structure of optimal paths. So the theoretical running time reduces to $\mathcal{O}(n \log n + m)$ and the practical runtime from about

one minute to few seconds on a road graph of Southern Germany with about 5 Mio. nodes and 11 Mio. edges. Finally, adapting the speed-up technique of *contraction hierarchies* [4] reduced the running time further to about 1/5th of a second, resulting in a total speedup of more than 300.

2.2 Taking into account Loading Stations

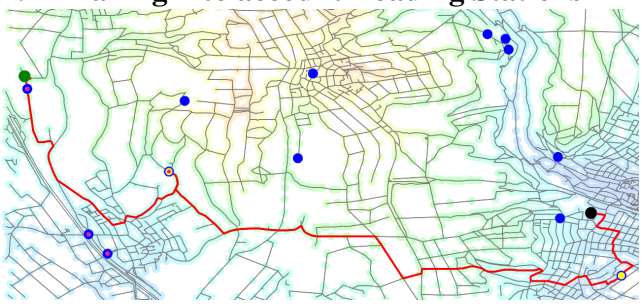

Figure 2: Path where recharging is necessary to reach the target (black) from the source (green). LSs are marked blue.

In spite of all energy optimization, the available battery load might simply not suffice to reach the desired target, hence the battery must be recharged or exchanged at loading stations (LS) on the way to complete the trip. Loading stations are represented as a set $L \subseteq V$ of distingushed nodes in the graph visiting whom recharges the battery to full capacity. As recharging is quite time consuming, a natural goal is to minimize the number of necessary recharging stops on the path. In [6] we solved this problem and presented a scheme whose basic building block is the computation of the *reachable set* $R(v)$ of a node v, i.e., all the nodes that can be reached from v under the energy constraints without recharging in between. Applying the techniques from [3] one can compute $R(v)$ in $\mathcal{O}(n \log n + m)$ time for a single $v \in V$. As a preprocessing step we construct an auxiliary graph $G_A(L, E_A)$ with $(l, l') \in E_A \Leftrightarrow l' \in R(l)$, reflecting the reachability between LSs. At query time it remains to compute $R(s)$ and the set of LSs t can be reached from (called $R^{-1}(t)$) which can be done similarly as $R(s)$. Then in G_A we determine the path with fewest recharging events via breadth-first-search started on the set of LSs $L \cap R(s)$ stopping as soon as we encounter some LS in $L \cap R^{-1}(t)$. This approach leads to query times below a second on the graph of Southern Germany.

2.3 Multi-criteria Objectives

It is unlikely that people are willing to accept considerably longer travel times just to save a few kWh of energy. The techniques presented so far do not guarantee anything in terms of travel time (or Euclidean distance). The following *multi-criteria* objectives are natural goals from a user perspective:

1. Find the energy-optimal path amongst all paths at most 10% longer/slower as the shortest/quickest path
2. Find the shortest/quickest energy-feasible path
3. Find the shortest/quickest feasible path with at most k recharging events
4. Find a feasible path with a minimal number of recharging events and bounded distance/travel time.

These problems at first sight look similar to the problem considered in Section 2.1 where we are to optimize some objective without violating some constraints (like the battery constraints). What turns these problems a lot more difficult, though, is the fact that now more than one metric is involved: apart from energy consumption, we are also interested in path length or travel time. In fact, all these problems are incarnations of the Constrained Shortest Path problem (CSP) which is NP-hard in general. For real-world problem instances of that type, though, we could show in [5] that an approach exploring all pareto-optimal solutions can be accelerated and improved such that queries even on large networks can be answered *optimally* within few seconds – something that was not possible before at all. The main technical contribution here was the extension of [4] such that pareto-optimal paths are preserved.

3. FACILITY LOCATION FOR LOADING STATIONS

With the limited cruising range being the biggest drawback of current EVs, the non-ubiquity of loading stations in most countries poses a severe problem to the widespread acceptance of E-mobility. So in particular in the early days of E-mobility it is of utmost importance to deliberately plan the locations of new loading stations for best possible coverage with EVs. The following (provably hard) algorithmic problems have been considered by us so far:

Going Anywhere without Getting Stuck: Where to place as few LSs as possible such that one can get from anywhere to anywhere with an EV. Unfortunately, this problem turns out to be closely related to Set-Cover and hence only approximable within $\mathcal{O}(\log n)$. Using instance-based lower bounds, good practical performance can be shown, though.

Make shortest/quickest Paths feasible: Ideally, we'd just plan our trip (disregards LSs) and simply rely on sufficiently many LSs being on the way (as it is the case with gas stations). We have come up with a Hitting-Set based formulation which due to the bounded VC-dimension of set systems of shortest paths can be approximated within a logarithmic factor of the optimum.

4. REFERENCES

[1] A.Artmeier, J.Haselmayr, M.Leucker, and M.Sachenbacher. The shortest path problem revisited: Optimal routing for electric vehicles. In *33rd Ann.German Conf. on Artif. Intell.(KI-2010)*, 2010.

[2] D.B.Johnson. Efficient algorithms for shortest paths in sparse networks. *J. ACM*, 24:1–13, January 1977.

[3] J.Eisner, S.Funke, and S.Storandt. Optimal route planning for electric vehicles in large networks. In *25th Conf. on Artificial Intelligence (AAAI)*, 2011.

[4] R.Geisberger, P.Sanders, D.Schultes, and D.Delling. Contraction hierarchies: faster and simpler hierarchical routing in road networks. In *Proc. 7th Int. Conf. on Experimental algorithms*, WEA'08, 2008.

[5] S.Storandt. Route planning for bicycles - exact constrained shortest paths made practical via contraction hierarchy. In *22nd Int. Conf. on Automated Planning and Scheduling (ICAPS)*, 2012.

[6] S.Storandt and S.Funke. Cruising with a battery-powered vehicle and not getting stranded. In *26th Conf. on Artificial Intelligence (AAAI)*, 2012.