Placement of Loading Stations for Electric Vehicles: No Detours Necessary!

University of Stuttgart
Germany

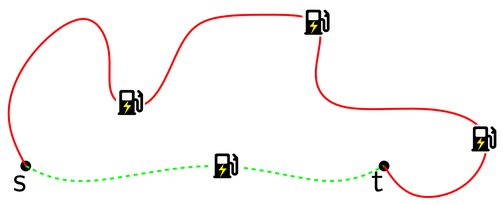
Stefan Funke, André Nusser **and** Sabine Storandt

UNI FREIBURG

ONE WAY

MOTIVATION

Electric vehicles have a limited cruising range ($\sim$ 125km) which together with the current sparsity of battery loading stations demands deliberate route planning and to put up with **detours**.



Goal: Place a *minimum number* of battery loading stations such that we can drive *every shortest path* without running out of energy!

MODELLING AS HITTING SET PROBLEM

Given:

- road network: graph $G(V, E)$
- edges have two metrics:
  - travel time
  - energy consumption

Assumptions:

- battery loading station can be positioned on any node
- full battery charge at the beginning of every trip

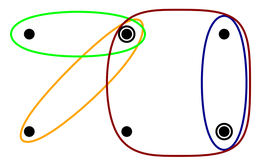
Approach: Formulate as *hitting set (HS)* problem $(\mathcal{U}, \mathcal{S})$ with

- $\mathcal{U}$ = set of nodes $= V$
- $\mathcal{S}$ = shortest paths where we run out of energy

. . . and solve via **Greedy Algorithm**

Recap:

- **Hitting Set Problem**: $(\mathcal{U}, \mathcal{S}), \mathcal{S} \subseteq 2^{\mathcal{U}}$
- Find minimum $H \subseteq \mathcal{U}$ such that $\forall s \in \mathcal{S} : s \cap H \neq \emptyset$.



This means we place at least one battery loading station on every shortest path we cannot drive without running out of energy.

PROBLEMS WITH THIS STRATEGY

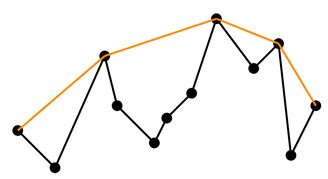
Extraction time and space for storing paths make this approach impractical!

For German road network ($|V| = 17.7M$):
time: $>$ **1 month**, space: $>$ **1000 terabyte**

- reduce size of individual sets by using more *space efficient path representations*
- adapt greedy hitting set algorithm to new path representations

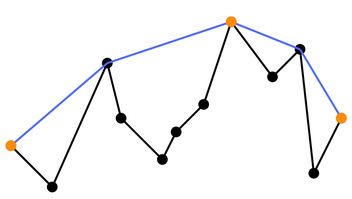
PATH EXTRACTION AND REPRESENTATION

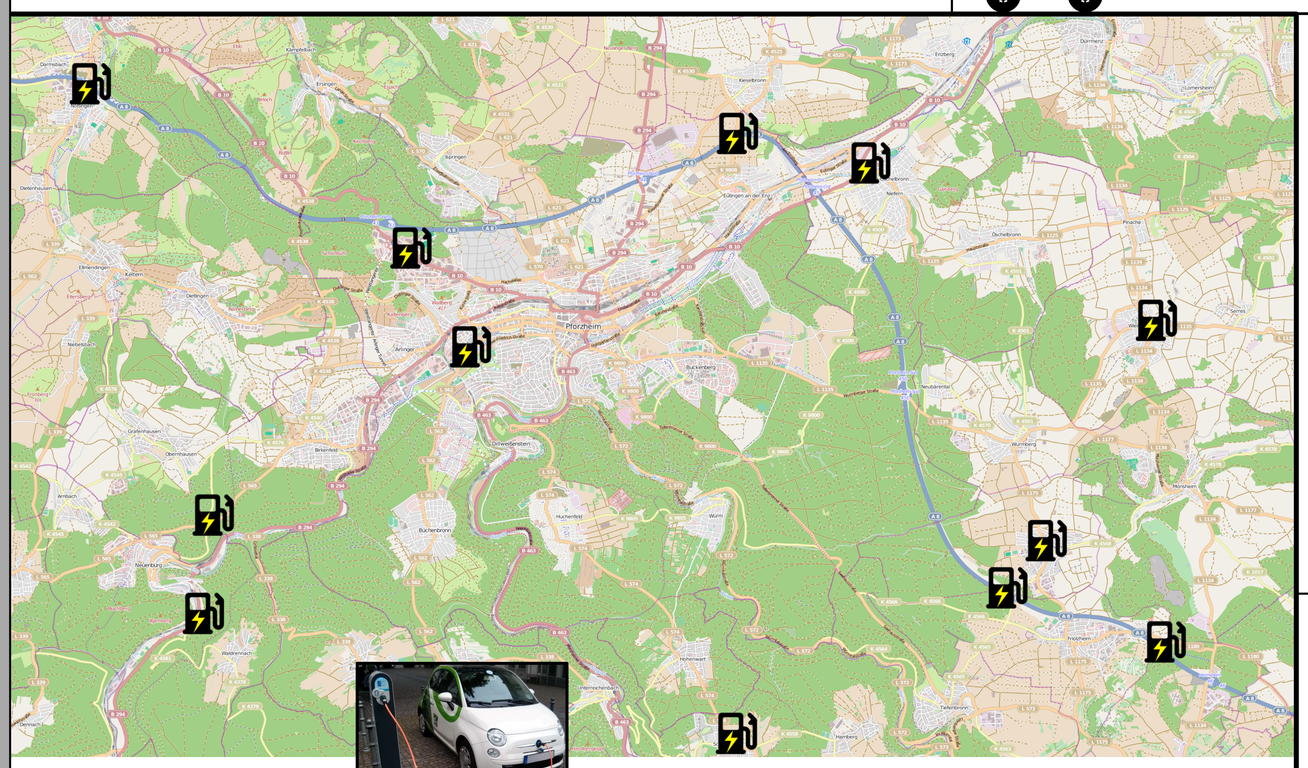
Contraction Hierarchies (CH) - based
→ store shortcut path



Dijkstra-based (Dij)
→ store vertex sets

Peak Node Mapping (PNM)
→ store (s, t, p)





battery loading station

a calculated placement of battery loading stations

FURTHER TECHNIQUES

- adapt greedy algorithm to path representations
- multiple hitters strategy
- multi-stage construction
- initial k-hop path cover

Setting:

- desktop hardware (i7-3930, 6 cores, 64GB RAM)
- road network of Baden-Württemberg ($|V| = 2.2M, |E| = 4.6M$)

Constructing the HS Instance (multicore)

rep.	time to construct HS instance (h:m)	space (GB)
Dij*	93:00	526
CH	33:00	34
PNM	00:47	14

* extrapolated values

FINAL RESULTS

Setting:

- desktop hardware (i7-2700, 4 cores, 32GB RAM)
- German road network ($|V| = 17.7M, |E| = 36M$)
- CH rep., all speed-ups, different multi-stage parameters
- cruising range: $\sim$ 125km

Solution Statistics (multicore):

comp. time (h:m)	#stations	guaranteed approximation factor a priori	a posteriori
5:22	728	16.7	< 3.83
3:29	1212	16.7	< 6.38