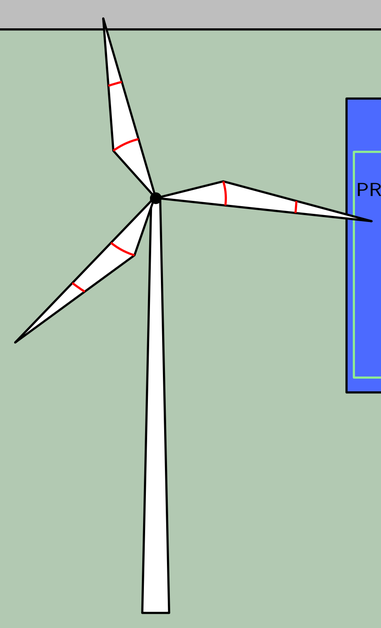
Enabling E-Mobility: One Way, Return, and with Loading Stations

Sabine Storandt **and** Jochen Eisner **and** Stefan Funke

University of Stuttgart
Germany



ELECTRIC VEHICLES (EVs)

PRO
- use regenerative energy
- recuperation capability
- low emissions
- low noise level

CON
- limited energy reservoir ⇒ limited cruising range
- long recharging time
- non-ubiquitous loading stations

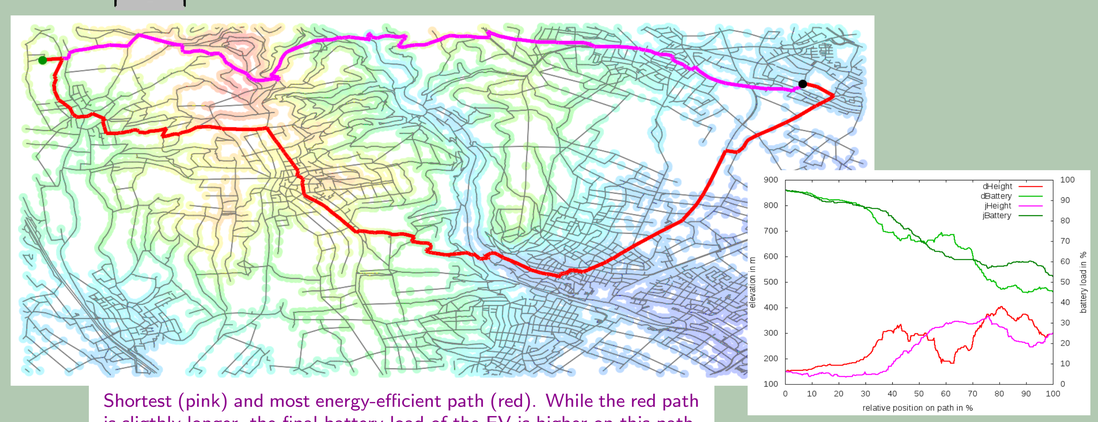
PLANNING ENERGY-EFFICIENT ROUTES

[Eisner, Funke, Storandt; AAAI 11]

Motivation

Choosing the most energy-efficient route (instead of the shortest one)

- is eco-friendly
- allows to exploit the whole cruising range
- might save recharging inbetween



Shortest (pink) and most energy-efficient path (red). While the red path is sligthly longer, the final battery load of the EV is higher on this path.

GIVEN
- street network $G(V, E)$, $|V| = n$, $|E| = m$
- costs $c : E \to \mathbb{R}$ representing energy consumption
- M maximal battery load of the electric vehicle (EV)

GOAL Compute path p from $s \in V$ to $t \in V$ with minimal energy consumption for initial battery load $I \leq M$.

PROBLEMS

1. partly negative edge eosts (due to the recuperation capability of EVs) ⇒ Dijkstra's algorithm can not be applied
2. battery constraints: battery load on every node in p must be ≥ 0 (no running out of energy) and $\leq M$ (no overcharging)

A path obeying the battery constraints is called **feasible**.

SOLUTION

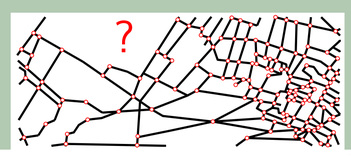
STEP 1: Model battery contraints into FIFO edge cost functions
Bellman-Ford applicable ⇒ query time $\mathcal{O}(nm)$

STEP 2: Apply Johnson's Shifting technique to get rid of negative edge costs
Dijkstra applicable ⇒ query time $\mathcal{O}(n \log n + m)$

STEP 3: Apply speed-up technique
Contraction Hierarchies (CH)

$n = 5.5$M, $m = 11$M	query time (sec)
Bellman-Ford	66.91
Dijkstra	3.43
CH	0.18

FACILITY LOCATION FOR LOADING STATIONS



Where to place loading stations reasonably?

Motivation
- dense network of loading stations (LSs) essential for driving longer tours
- LSs are still sparse in many areas
- important to plan the locations of new LSs for best possible coverage

GOAL Place as few loading stations as possible such that...

... one can drive from anywhere to anywhere (ERC)
- closely related to SetCover
- not approximable in $c \cdot \ln n$ for $c < 1$

... one can drive to any destination and back (ECC)
- closely related to SetCover by Pairs
- even worse inapproximability bound

... all shortest paths are feasible
- closely related to HittingSet
- systems of shortest paths have bounded VC-dimension ⇒ $\log(OPT)$ APX available

good heuristics available (e.g. greedy)
quality provable via instance-based lower bounds

avg. cruising range	lower bound	ERC	ECC
75 km	105	339	812
100 km	49	147	379
125 km	37	103	268
150 km	22	61	187
175 km	14	46	138

Instance-based lower bounds and heuristic results for the number of LSs necessary to achieve the goals ERC and ECC in the graph of Germany.

OVERALL GOAL
develop algorithmic tools to solve fundamental problems in E-Mobility in order to make EVs competetive to conventional cars

TAKING INTO ACCOUNT LOADING STATIONS

[Storandt, Funke; AAAI 12]

A **Loading Station (LS)** is a node $l \in V$, that leads to a full battery ($b_l = M$), whenever it is visited.

The set of all LSs is called $L \subseteq V$.

GOAL Compute path from $s \in V$ to $t \in V$ with minimal number of necessary recharging events.

APPROACH Build auxiliary graph Q upon loading stations in a preprocessing step.

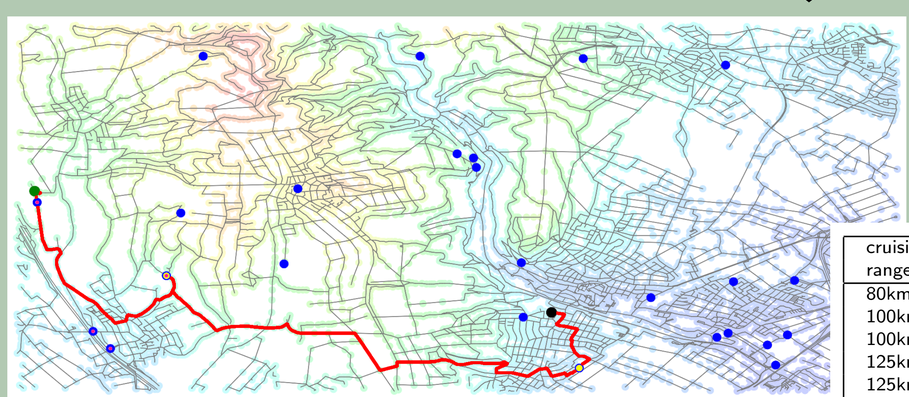
$Q(L, E')$
- L set of LSs
- $(l, l') \in E'$ $(l, l' \in L)$ if l' is *reachable* from l

A node $w \in V$ is **reachable** from $v \in V$, if there exists a path from v to w which obeys the battery constraints and requires no recharging inbetween.

QUERY ANSWERING

1. Compute $L_s \subseteq L$ set of reachable LSs from s. time $\mathcal{O}(n \log n + m)$
2. Compute $L_t \subseteq L$ set of LSs t can be reached from. time $\mathcal{O}(n \log n + m)$
3. Run BFS from L_s in Q until the first node in L_t is visited. time $\mathcal{O}(n)$
4. Backtrack respective path. time $\mathcal{O}(n)$

Total run time $\mathcal{O}(n \log n + m)$



Path where recharging is necessary twice to reach the target (black) from the source (green). Loading stations are marked blue.

$n = 5.5$M, $m = 11$M

cruising range	# LS	query time (sec)
80km	200	0.17
100km	100	0.32
100km	200	0.32
125km	50	0.47
125km	100	0.45
150km	50	0.53
150km	100	0.51

MULTI-CRITERIA OBJECTIVES

Natural Goals

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.

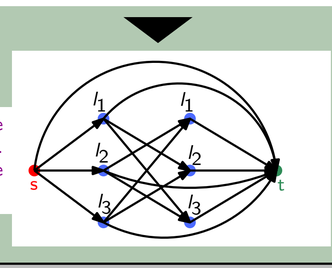
Instances of the Constrained Shortest Path (CSP) problem
NP-hard in general

Approach for 1. & 2.
Label setting computation (LSC) still works if cost or resource consumption are functions!

Adapt speed-up technique Contraction Hierachies for CSP to decrease query times for LSC.

Approach for 3. & 4.
Shortest feasible paths between loading stations are invariant under the choice of s, t and l.
⇒ precomputation possible, build layered graph

Augment layered graph on query time with proper edges from s and to t (retrieved by two local LSCs)
⇒ single Dijkstra run reveals the optimal path



Layered graph to find the shortest feasible path with at most two recharging events. Possible loading stations are marked blue (l_1, l_2, l_3).