An Efficient Electric Vehicle Path-Planner That Considers the Waiting Time

Using graph relabeling and alternative paths generation

Jaël Champagne Gareau
Éric Beaudry
Vladimir Makarenkov

Computer Science Department
Université du Québec à Montréal

November 8th, 2019

Outline

Advantages of EV over conventional vehicles

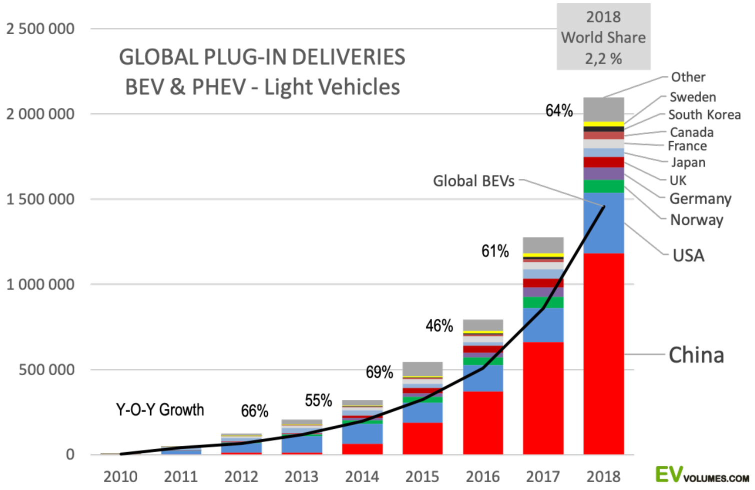
Less pollution

Less noisy

Cheaper in the long run

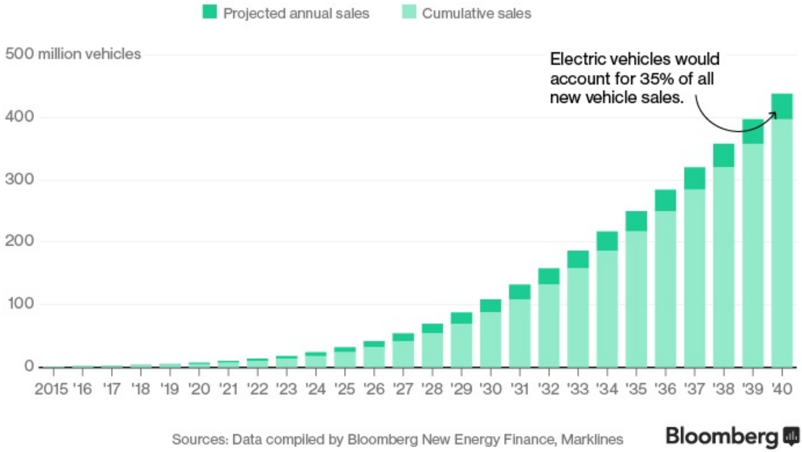
Less maintenance

Global EV market 2010–2018[1]



[1] http://www.ev-volumes.com/country/total-world-plug-in-vehicle-volumes/

EV sales forecast[2]



[2] Bloomberg, February 25th, 2016, https://www.bloomberg.com/features/2016-ev-oil-crisis/

Comparison between a conventional vehicle and an EV

	Honda Civic	Nissan Leaf
Price (C$)[3]	17 890 $	42 298 $
Range	750 km	363 km
Refueling/Charging time	3 min	30 min
Gas/L3 Charging stations[4]	2924	225

[3] Starting price for the 2019 model. Excluding governmental subsidies for green vehicles

[4] In Québec province, Canada, in 2018

Research problem

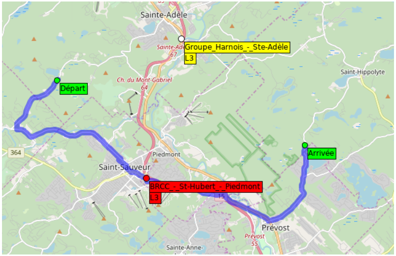
- The number of EV is increasing;
- Many paths need recharges to be feasible.

Objective

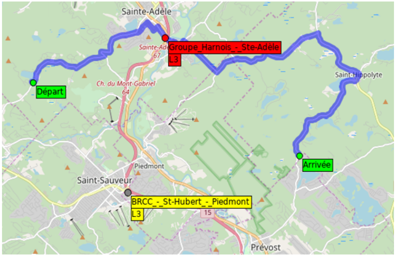
The objective is to have an EV planner that:

1. considers intermediate recharges at charging stations;
2. **considers the expected occupancy and waiting time at the stations.**

Real world example of the impact of the consideration of waiting time



Monday noon



Tuesday noon

Related Work

EVRP-MRUA[5]

- EV Routing Problem with Mid-Route Recharging and Uncertain Availability.
- Deliver mulitple packages in an optimal order (similar to the TSP).
- Minimize the cost for the operator and the global time to deliver the packages.

[5]Nicholas Kullman, Justin Goodson, Jorge E. Mendoza. Dynamic Electric Vehicle Routing with Mid-route Recharging and Uncertain Availability. ODYSSEUS 2018, Jun 2018, Cagliari, Italy.

Related Work

EVRP-MRUA[5]

- EV Routing Problem with Mid-Route Recharging and Uncertain Availability.
- Deliver mulitpWhile packages in an optimal order (similar to the TSP).
- Minimize the cost for the operator and the global time to deliver the packages.

Adaptive Routing and Recharging Policies for EV[6]

- Single EV going from a departure to an arrival node.
- Considers expected waiting time and EV stations availability uncertainty.
- Availability of EV station assumed to be known only when arriving.
- Every node is a station. Time complexity of $\mathcal{O}(n^4)$.

[5] Nicholas Kullman, Justin Goodson, Jorge E. Mendoza. Dynamic Electric Vehicle Routing with Mid-route Recharging and Uncertain Availability. ODYSSEUS 2018, Jun 2018, Cagliari, Italy.

[6] Timothy M Sweda, Irina S Dolinskaya, and Diego Klabjan. 2017. Adaptive Routing and Recharging Policies for Electric Vehicles. Transportation Science 51, 4 (2017), 1326–1348.

Formalism

Road Network

The **network** is a tuple (V, E, λ, μ, S), where (V, E) is a digraph. More specifically:

- V is the set of locations considered on the map (nodes);
- E is the set of road segments (edges);
- $\lambda\colon E \to \mathbb{R}^+$ gives the length (in m) of the edges;
- $\mu\colon E \to \mathbb{R}^+$ gives the expected speed (in m/s) at the edges;
- S is the set of charging stations (we assume that $S \subseteq V$).

Formalism

Road Network

The **network** is a tuple (V, E, λ, μ, S), where (V, E) is a digraph. More specifically:

- V is the set of locations considered on the map (nodes);
- E is the set of road segments (edges);
- $\lambda\colon E \to \mathbb{R}^+$ gives the length (in m) of the edges;
- $\mu\colon E \to \mathbb{R}^+$ gives the expected speed (in m/s) at the edges;
- S is the set of charging stations (we assume that $S \subseteq V$).

EV Planning Problem (EVPP)

An **EVPP** is defined by a tuple $(M, \rho, \alpha, \omega)$, where

- M is the road map;
- $\rho \in \mathbb{R}^+$ is the EV range;
- $\alpha, \omega \in V$ are the departure and arrival nodes.

EVPP Solution

Solution

A **solution** to an EVPP $(M, \rho, \alpha, \omega)$ is a tuple (P, Q), where

- $P \subseteq V$ is the sequence of nodes to traverse in the solution;
- $Q \subseteq P$ contains the stations where to charge (and α, ω);
- $\forall i, \mathrm{dg}(Q_i, Q_{i+1}) \leq \rho$, where dg is the graph distance.

EVPP Solution

Solution

A **solution** to an EVPP $(M, \rho, \alpha, \omega)$ is a tuple (P, Q), where

- $P \subseteq V$ is the sequence of nodes to traverse in the solution;
- $Q \subseteq P$ contains the stations where to charge (and α, ω);
- $\forall i, \mathrm{dg}(Q_i, Q_{i+1}) \leq \rho$, where dg is the graph distance.

Optimal Solution

An **optimal solution** is a solution (P, Q) minimizing:

$$Z(P, Q) = \mathrm{DT}(P) + \mathrm{CT}(Q) + \mathrm{WT}(Q),$$

where DT, CT and WT are the expected driving, charging and waiting time.

Base algorithm

Algorithm Base planner

```
Compute the matrix D and the optimal path between every pair of stations
Construct the s-graph containing every charging station
for each request (α, ω, ρ) do
    Run Dijkstra from α on the original graph
    Run Dijkstra from ω on the reversed original graph
    Add α and ω to the s-graph and add edges with length ≤ ρ
    Run the A* algorithm on the s-graph from α to ω to find the sequence Q
    Find the sequence P from Q using all computed paths
end for
```

The time complexity for each request is $\mathcal{O}(|V| \log |V| + |E|)$.
This planner can be extended to:

- Consider partial initial EV charge;
- Consider the regenerative braking of EVs;
- Consider partial recharge and the non-linear charging curve.

The idea

- Consider historic data of stations occupancy;
- Relabel the graph to account for these data.

A priori known data

For every station s, the time-dependent probability of occupancy is given by:

$$\begin{aligned} f_s \colon \{Monday, \ldots, Sunday\} \times \{0..23\} &\to [0, 1] \\ (d, h) &\mapsto \mathbb{P}(s \text{ is occupied} \mid \text{Day} = d \wedge \text{Hour} = h). \end{aligned}$$

and the time-dependent expected waiting-time when occupied is given by:

$$g_s \colon \{Monday, \ldots, Sunday\} \times \{0..23\} \to \mathbb{R}^+$$

Time-dependent graph relabeling

Labeling considering the waiting time

Let $e = (u, v) \in E$. We define the time-dependent labeling to be

$$\xi \colon E \times \{Monday, \ldots, Sunday\} \times \{0..23\} \to \mathbb{R}^+$$

$$\xi(e, d, h) = \begin{cases} \lambda(e) & \text{if } u \notin S \\ \lambda(e) + f_u(d, h) \cdot g_u(d, h) \cdot \mu(e) & \text{if } u \in S \end{cases}$$

- The edge weight now depends on the time of arrival;
- We need to modify the graph search algorithm (e.g., Dijkstra/A*)[7];

[7] Daniel Delling and Dorothea Wagner. 2009. Time-dependent route planning. In Lecture Notes in Computer Science, Vol. 5868 LNCS. 207–230. https://doi.org/10. 1007/978-3-642-05465-5_8

Problem with previous technique

- The previous technique uses only *a priori* known data;
- Real-time occupancy can be much worse than what was expected;
- Assume we have access to real-time occupancy while driving;
- We can further reduce the waiting time by precomputing alternative paths.

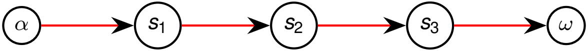
Two extreme cases

1. No alternative path;
2. A total policy π: State $\rightarrow$ Action (e.g., found using MDP)

We want a compromise between these two extreme cases.

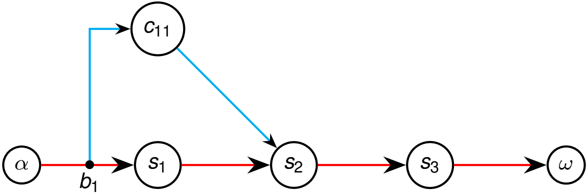
The idea

Generate one alternative path for every station on the initial path.



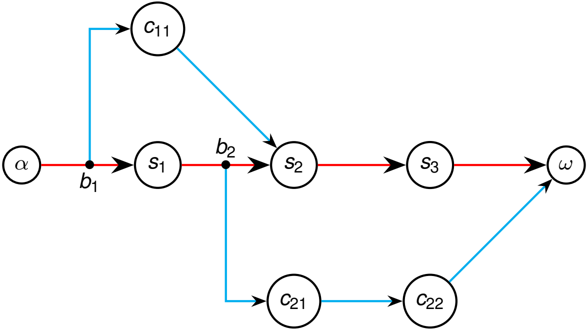
The idea

Generate one alternative path for every station on the initial path.



The idea

Generate one alternative path for every station on the initial path.



Generating the alternative paths

Algorithm Alternative path generation for station s_i

1: Assume $f_{s_i} \equiv 1$
2: Run the relabeled time-dependent planner
3: **if** new path is same as base path **then**
4:     **return**
5: **end if**
6: $b_i \leftarrow$ last common node in prefix of new path and base path
7: Set the new path as an alternative path on node b_i

After running this algorithm on every station, we obtain a partial policy

$$\pi : V \to V^2$$

$$\pi(x) = \begin{cases} (s_{i+1}, -) & \text{if } x = s_i \wedge \nexists b_{i+1} \\ (b_i, -) & \text{if } x = s_i \wedge \exists b_{i+1} \\ (s_i, c_{i1}) & \text{if } x = b_i \\ (c_{i,j+1}, -) & \text{if } x = c_{ij}, \end{cases}$$

Executing the policy

Algorithm Online plan execution

```
procedure EXECUTEPLAN(π)
    n ← α
    while n ≠ ω do
        (x, y) ← π(n)
        if y = − ∨ ¬occupied(x) then
            n ← x
        else
            n ← y
        end if
        Move EV to node n
    end while
end procedure
```

Test methodology

- The real map data come from the OpenStreetMap project.
- The territory used is the Province of Québec, Canada:
  - 2 923 013 nodes
  - 5 907 672 edges
- The charging stations data come from the Circuit Électrique:
  - 1318 charging stations (1178 L2 and 140 L3)
  - the f_s and g_s functions were generated from the data.
- 1000 requests:
  - EV range was generated uniformly between 90 and 550 km;
  - α and ω were chosen at random among all nodes;
  - Travel distance was between 200 and 1500 km.

Results: Table showing the results obtained for real data

Parameters		Baseline		Relabeling			Alternative Paths		
SN	PM	WT min	TT min	WT min	WR %	TTR min	WT min	WR %	TTR min
R140	× 1	10.2	350.1	3.3	-67.5	-6.2	2.2	-78.8	-7.2
R140	× 2	22.7	362.5	6.0	-73.6	-15.5	4.1	-82.0	-17.2
R140	× 3	37.5	377.3	7.7	-79.3	-28.3	7.2	-80.9	-28.7
R140	Rand	51.3	391.2	10.9	-78.7	-38.5	9.8	-80.9	-39.5
R1318	× 1	19.4	356.0	2.2	-88.6	-16.7	1.7	-91.5	-17.3
R1318	× 2	37.5	374.0	4.0	-89.3	-32.8	3.0	-92.0	-33.7
R1318	× 3	50.5	387.1	6.5	-87.1	-43.1	4.7	-90.6	-44.8
R1318	Rand	62.0	398.6	6.9	-88.8	-53.3	6.0	-90.4	-54.2

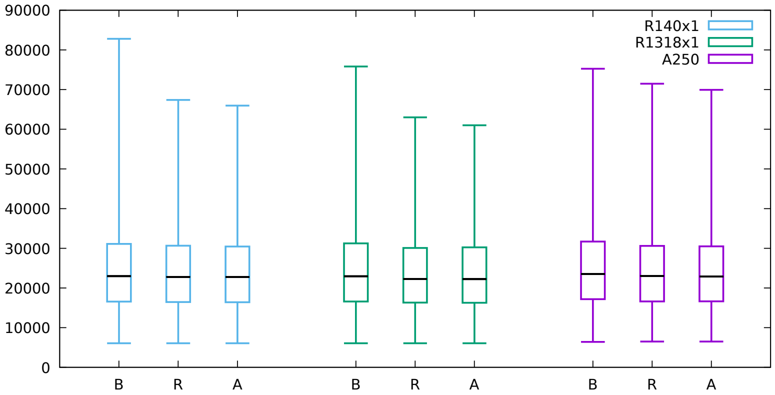
SN: Station Network; **PM**: Probability Modifier;**TTR**: Total time reduction (vs Baseline)

Results: Table showing the results obtained for artificial data

Parameters		Baseline		Relabeling			Alternative Paths		
SN	PM	WT min	TT min	WT min	WR %	TTR min	WT min	WR %	TTR min
A250	Rand	29.3	368.2	12.0	-59.0	-13.9	10.3	-64.8	-15.1
A500	Rand	28.9	363.4	9.4	-67.6	-16.5	8.3	-71.2	-17.3
A1000	Rand	28.7	362.5	7.4	-74.2	-18.7	6.5	-77.3	-19.4
A2000	Rand	27.1	359.9	4.9	-81.9	-19.9	3.8	-86.0	-20.9

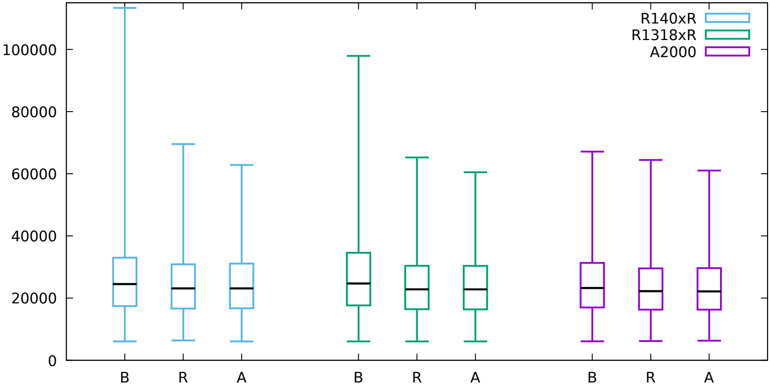
SN: Station Network; **PM**: Probability Modifier;**TTR**: Total time reduction (vs Baseline)

Results: Box plots showing the five number summary of the total time



B: Baseline; **R**: Relabeling; **A**: Alternative paths.

Results: Box plots showing the five number summary of the total time



B: Baseline; **R**: Relabeling; **A**: Alternative paths.

Conclusion

- The waiting time can have a significant impact in EV planning.
- We proposed two techniques:
  - a dynamic time-dependent graph relabeling;
  - an alternative path generation mechanism to account for worse than expected occupancy.
- Both techniques have a negligible computation overhead over the base planner.
- Our techniques decreased by more than 3/4 the waiting time in our simulations, representing a 17.3 minutes saving on average.

Acknowledgements

We acknowledge the support of the Natural Sciences and Engineering Council of Canada (NSERC) and the Fonds de recherche du Québec — Nature et technologies (FRQNT).