Cooperative Electric Vehicles Planning

Jaël Champagne Gareau
Marc-André Lavoie
Guillaume Gosset
Éric Beaudry

Computer Science Department
Université du Québec à Montréal

6 – 10 may 2024

UQÀM

AAMAS 2024 AUCKLAND NZ

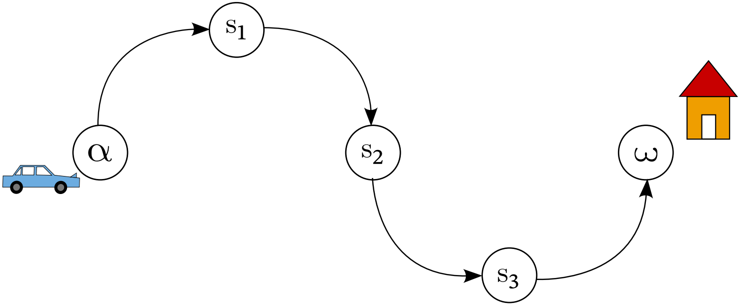
Outline

Electric Vehicles Planning

- EVs are becoming increasingly widespread due to :
  - environmental concerns ;
  - improvements in their battery range ;
  - increased charging stations availability.
- There are some challenges specific to EV planners, e.g., :
  - intermediate stops for recharging when the journey is too long ;
  - unpredictable waiting times at the charging stations ;
  - regenerative braking.

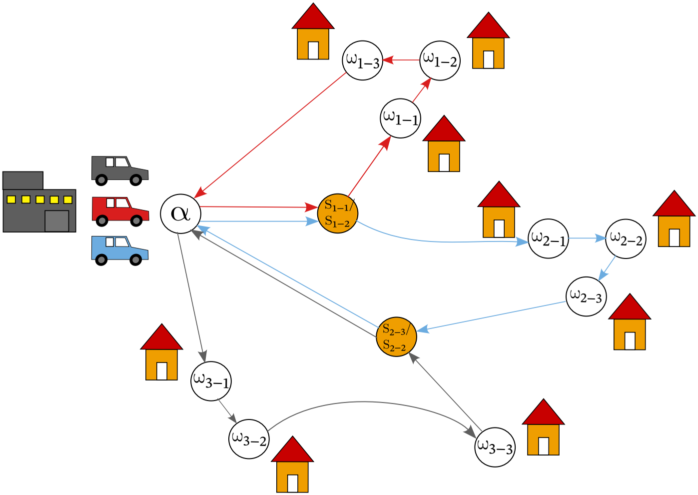
Electric Vehicles Path-Planning (EVPP)

- Single EV path-planning from α to ω in a road network ;
- The EV has a range ρ and must hop from stations to stations ;
- Many variants (consideration of regenerative braking, waiting times, etc.)



Electric Vehicles Routing Problem (EVRP)

- A fleet of EVs controlled by the same entity and sharing the same objective ;
  - E.g., deliver packages from a depot/warehouse to a set of locations ;
- Goal : find a mininum set of EVs able to complete all tasks with minimal cost ;

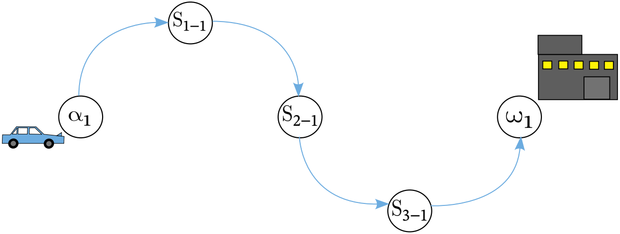


Motivation

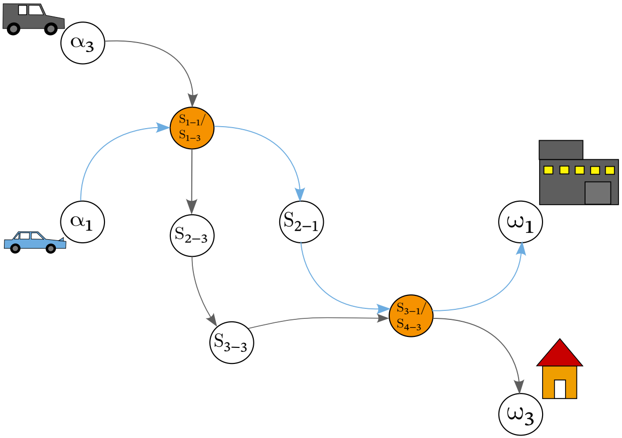
"An open challenge is to devise algorithms for socially optimal real-time routing with a reasonable response time for a large number of vehicles."[1]

1. Basharzad, S. N., Choudhury, F. M., Tanin, E., Andrew, L. L. H., Samet, H., & Sarvi, M. (2022). Electric vehicle charging : It is not as simple as charging a smartphone. Proceedings of the 30th International Conference on Advances in Geographic Information Systems, 1–4. https ://doi.org/10.1145/3557915.3560967

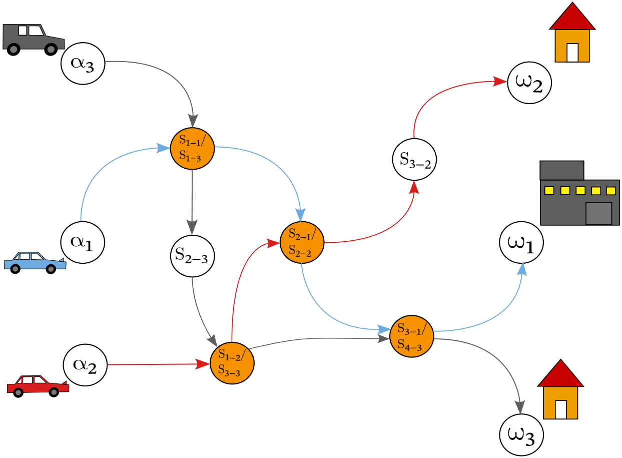
Motivation – Example



Motivation – Example



Motivation – Example



Cooperative Electric Vehicles Planning Problem (CEVPP)

- There are many EVs, controlled by different end-users, each with their own goal.
- It is desirable to plan their routes collectively to reduce global waiting times.
- EV drivers can send a planning request to a centralized planner.
- New EVs can enter the planning problem at any time.
- In practice, the planner can recompute a global plan
  - every N new requests to the planner since the last replanning ;
  - every T minutes.
- In this research, we focus on a batch of EV requests during a given replanning.

Main differences between EVRP and CEVPP

EVRP	CEVPP
EVs start and end at same position	Each EV has its own start and end
The EVs cooperate to reach a common goal	Each EV has its own goal
The problem is static / offline	The problem is dynamic / online
Find min-set of EVs able to complete all tasks with min-cost	Minimize the global plan cost (travel + charging + waiting) times

CEVPP – Definition

Road Network

We define a **road network** M as a tuple (V, E, λ, μ, S), where :

- V is the set of nodes (latitude, longitude) on the map ;
- E is the set of road segments (edges) ;
- $\lambda \colon E \to \mathbb{R}^+$ gives the length (in m) of every edge ;
- $\mu \colon E \to \mathbb{R}^+$ gives the expected speed (in m/s) at every edge ;
- $S \subseteq V$ is the set of all charging stations.

EV Request

Each EV has an associated **EV request**, i.e., a tuple $(\alpha, \omega, \rho, \tau)$, where :

- α is the departure node ;
- ω is the arrival node ;
- ρ is the range of the EV ;
- τ is the time of departure.

CEVPP – Definition

CEVPP instance

A **CEVPP instance** is a tuple (M, R) where :

- M is a road network ;
- $R = \langle(\alpha_1, \omega_1, \tau_1, \rho_1), \ldots, (\alpha_k, \omega_k, \tau_k, \rho_k)\rangle$ is a list of EV requests in an arbitrary order.

Objective

The **objective** of a CEVPP instance is to find a solution $\pi = \langle \pi_1, \pi_1, \ldots, \pi_k \rangle$ that

- minimizes total (travel + charge + wait) time of the batch of EVs.
- $\pi^\star = \arg\min_{\pi \in \Pi} \left[\frac{1}{k} \sum_{i=1}^{k} \left(C(\pi_i) - C^\star(\pi_i) \right)^2 \right]$.
- $C^\star(\pi_i)$ is the cost of the optimal plan of the i^{th} EV when it is alone in M, i.e., :
  - geographically the shortest-path ;
  - no waiting time.

Baseline planner

- We precompute a **stations' graph** $G = (S, E')$ with the Floyd-Warshall algorithm.
- We assume, without loss of generality, that $(\alpha, \omega) \in S^2$ are in G

Algorithm Baseline Non-Cooperative EV Planner

```
procedure NCEVP((M, R = ⟨r_1, ..., r_k⟩) : CEVPP, G : stations' graph)
    for all r_i ∈ R do
        ▷ Considers travel and charging, but not waiting time
        π_i ← A*(M, r_i)                ▷ Only considers edges e with length λ(e) < ρ_i
        π ← π ∪ {π_i}
    Compute the global penalty P(π)     ▷ Entirely due to waiting times
```

- Time complexity of NCEVP : $\Theta(k \cdot |S|^2)$.

Optimal Planner

- We propose an optimal planner that uses a graph-planning algorithm to search in a graph representing the problem's state-space.

State

We define a **state** to be an array $\sigma = [(\sigma_1^s, \sigma_1^t), (\sigma_2^s, \sigma_2^t), \ldots, (\sigma_k^s, \sigma_k^t)]$, where :

- σ_i^s is the charging station currently used by the i^{th} EV ;
- σ_i^t is the planned departure time of the i^{th} EV from station σ_i^s.

Optimal Planner

Algorithm Exhaustive-Search Cooperative EV Planner

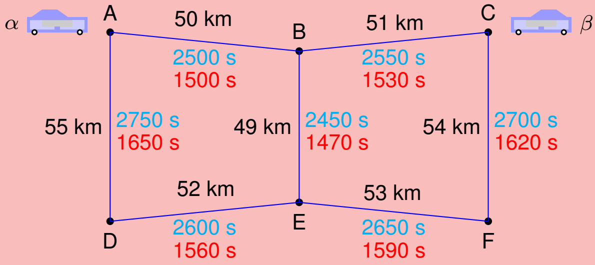
procedure $\textsc{escEVP}((M, R = \langle r_1, \ldots, r_k \rangle) : \text{CEVPP})$
    open $\leftarrow$ Empty Priority Queue of (state, cost $f = g + h$)
    open.push$(\textsc{InitialState}(M, R), 0)$
    while not *open.empty*() **do**
        $\sigma \leftarrow$ *open.pop*()
        if $\textsc{IsGoalState}(\sigma)$ **then** $\sigma^\star \leftarrow \sigma$; break
        for all vehicle $i \in \{1, \ldots, k\}$ **do** ▷ any EV can move
            for all $s \in \textsc{ReachableStations}(\sigma_i^s, \rho_i)$ **do**
                if i^{th} EV already visited s **then** continue
                $\sigma' \leftarrow \sigma$ ▷ state σ' is same as σ except for the i^{th} EV
                $\sigma'[i] \leftarrow (s, \textsc{ComputeTimeDeparture}(i, s, \sigma))$
                $f \leftarrow \min_{i \in \{1, \ldots, k\}} (\textsc{Cost}(i, \sigma') + \textsc{Heuristic}(i, \sigma', r_i))$
                open.push(σ', f)
    Extract global plan π from $\sigma^\star$
    Compute the global penalty $P(\pi)$

- Time complexity of $\textsc{escEVP}$: $\Omega(|S|^k)$.

Permutations Planner

- We propose another cooperative planner, inspired by Cooperative-A*.
- It computes a plan for each EV one-by-one, but records charging stations occupancy in a reservation table.
- The Modified-A* algorithm considers the waiting time due to existing reservations when planning a new EV.

Non-optimality of the solution



$\langle \alpha, \beta \rangle \rightarrow$ 13670 s $\quad \langle \beta, \alpha \rangle \rightarrow$ 13610 s $\quad \pi^* \rightarrow$ 12720 s

Permutations Planner

Algorithm Permutations Cooperative EV Planner

procedure PCEVP($(M, R = \langle r_1, \ldots, r_k \rangle)$: CEVPP)
  $\mathcal{P} \leftarrow$ GETCONSIDEREDPERMUTATIONS(R)
  $C_{best} \leftarrow \infty$
  for all $\phi \in \mathcal{P}$ **do**
    $\pi \leftarrow \emptyset$
    $\mathcal{R} \leftarrow$ Empty Reservation Table
    for all $r_i \in \phi$ **do** ▷ In given order
      $\pi_i =$ MODIFIEDA*$(M, r_i, \mathcal{R})$
      UPDATERESERVATIONTABLE$(\mathcal{R}, \pi)$
      $\pi \leftarrow \pi \cup \{\pi_i\}$
    if $C(\pi) < C_{best}$ **then**
      $\pi_{best} \leftarrow \pi$
      $C_{best} \leftarrow C(\pi)$
  Compute the global penalty $P(\pi_{best})$

- Time complexity : $\Theta(|\mathcal{P}| \cdot |S|^2)$.

Methodology

- We compared the baseline planner to three different instances of pcEVP :
  - only one permutation, where EVs are ordered by time of departure τ ($\Theta(|S|^2)$) ;
  - random $\log(k!)$ permutations ($\Theta(k \log k \cdot |S|^2)$;
  - cascade permutations ($\Theta(k^2 \cdot |S|^2)$.
- Empirical evaluation is done on two regions of Canada (OpenStreetMap) :
  - Maritimes (2 105 607 vertices and 4 200 189 edges) ;
  - Québec (4 416 080 vertices and 8 797 051 edges).
- We used real charging stations data from the *Electric Circuit*.
  - Maritimes had 50 charging stations ;
  - Québec had two tested subset of stations (347 and 1816 stations).
- All algorithms were implemented in C++ and compiled with g++ (version 12.2).
- Experiments were performed on a 4.2 GHz Intel Core i5-7600k CPU.
- We measured two metrics :
  - running time of the algorithms ;
  - penalty $\frac{1}{k}\sum_{i=1}^{k}(C(\pi_i) - C^\star(\pi_i))^2$ of the solutions.
- EV requests :
  - Range ρ is sampled uniformly between 100 and 550 km.
  - Departure time τ is sampled uniformly between 0 and 120 minutes.
  - The departure α (resp. arrival ω) of each EV is sampled from a 50 km cluster.
- We used a timeout value of 15 minutes per request.

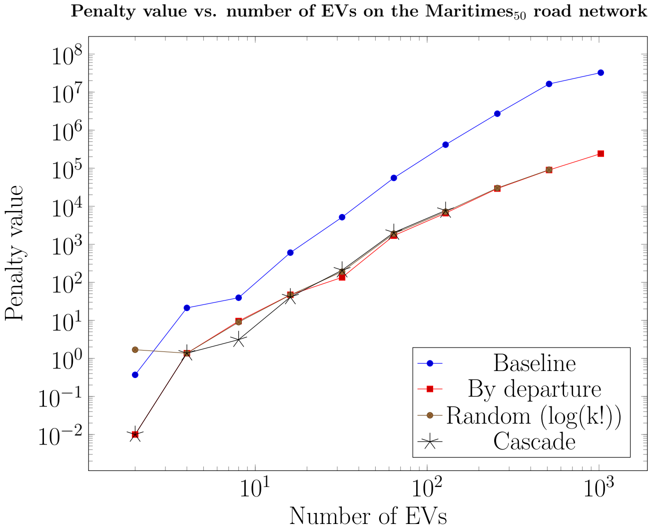
Average running times (ms)

Network	**Baseline**	**By departure**	**Random** $\log(k!)$	**Cascade**
Maritimes_{50}	0.09	0.19	95.35	1459.2
Quebec_{347}	2.272	2.70	99.27	558.86
Quebec_{1816}	93.84	103.76	1058.18	3656.6
Average	32.07	35.55	417.60	1891.55

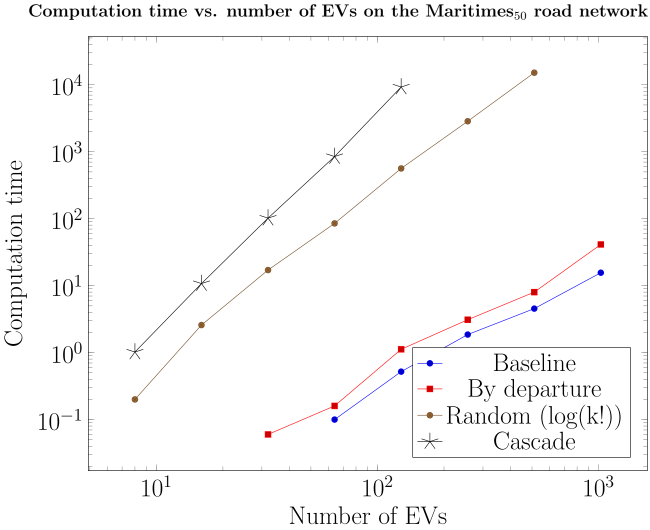
Average reduction (%) in penalty (min) compared to baseline

Network	**By departure**	**Random** $\log(k!)$	**Cascade**
Maritimes_{50}	93.06	93.07	95.22
Quebec_{347}	86.33	86.73	89.35
Quebec_{1816}	96.69	97.57	98.25
Average	92.03	92.46	94.27

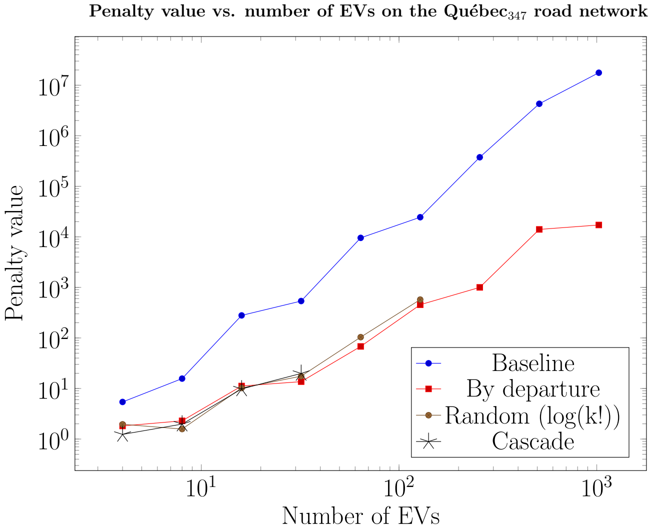
Penalty on $Maritimes_{50}$



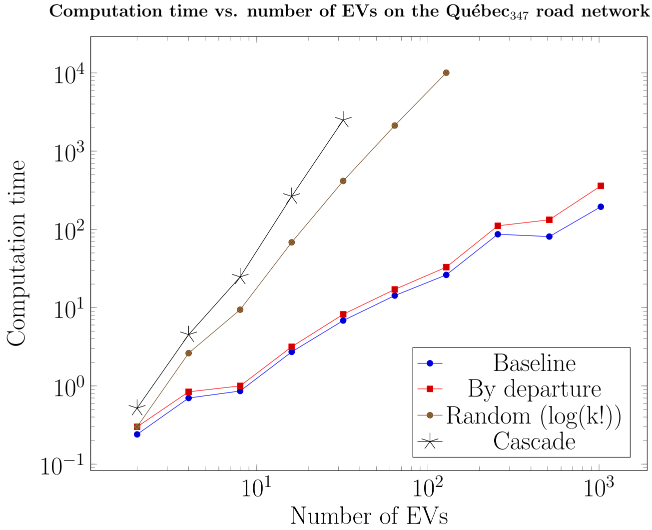
Computation times on $Maritimes_{50}$



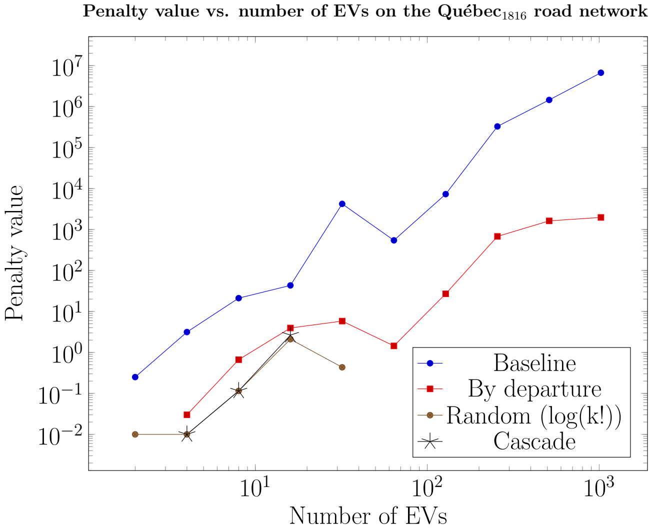
Penalty on Quebec_{347}



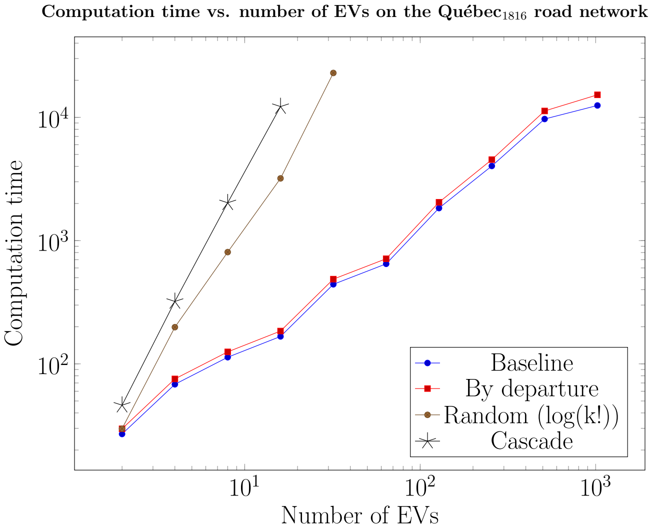
Computation times on Quebec$_{347}$



Penalty on Quebec$_{1816}$



Computation times on Quebec$_{1816}$



Conclusion

- We introduced the new CEVPP problem.
- Overall time can drastically be reduced (2h per EV, on average).
- As the number of EVs grows, the number of bottlenecks at stations grows too, presenting more opportunities for optimization and further emphasizing the relevance of CEVPP.
- Future works :
  - Finding ways of pruning large part of the state-space, to make that optimal planner more useful for real-world applications.
  - Conduct a comprehensive analysis of various permutation subsets.
  - Consider waiting times caused by EVs external to our planner.

Acknowledgments

CRSNG NSERC

Fonds de recherche Nature et technologies Québec

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).