Planification d'itinéraires pour véhicules électriques

Considération du temps d'attente et regroupement de bornes

Jaël Champagne Gareau

Séminaire INF889B
Département d'informatique
Université du Québec à Montréal

5 février 2020

Plan de la présentation

UQÀM

Pourquoi des véhicules électriques ?

Moins de GES

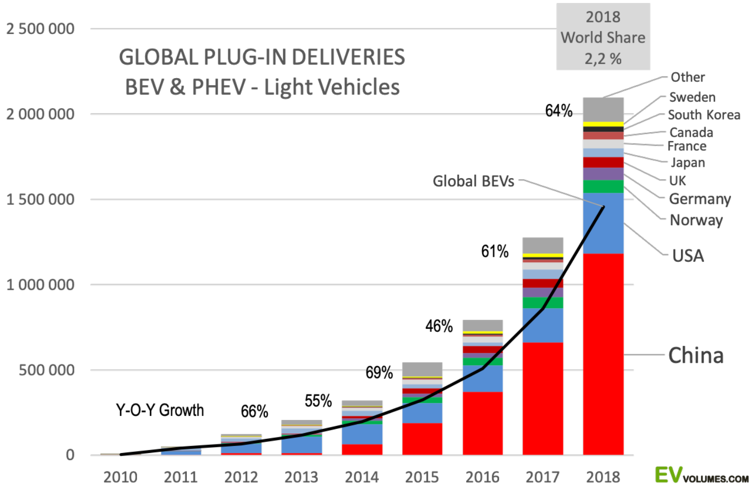
Moins bruyant

Plus économique à long terme

Moins d'entretien

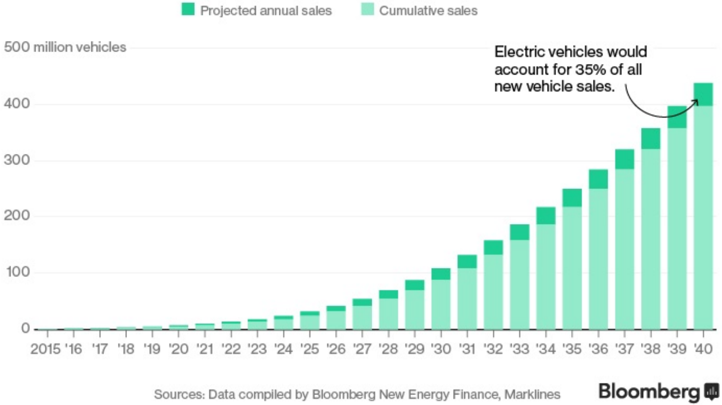
UQÀM

Marché mondial des VÉ 2010–2018 [1]



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

Prédiction de l'évolution des ventes de VÉ [2]



2. Bloomberg, 25 février 2016, https ://www.bloomberg.com/features/2016-ev-oil-crisis/

Comparaison entre un véhicule conventionnel et un VÉ

	Honda Civic	Nissan Leaf
Prix (C$) [3]	17 890 $	42 298 $
Autonomie	750 km	363 km
Temps de recharge	3 min	30 min
Stations service / Bornes de recharge [4]	2924	225

3. Modèle 2019, excluant les subventions gouvernementales
4. Au Québec en 2019

Problématique

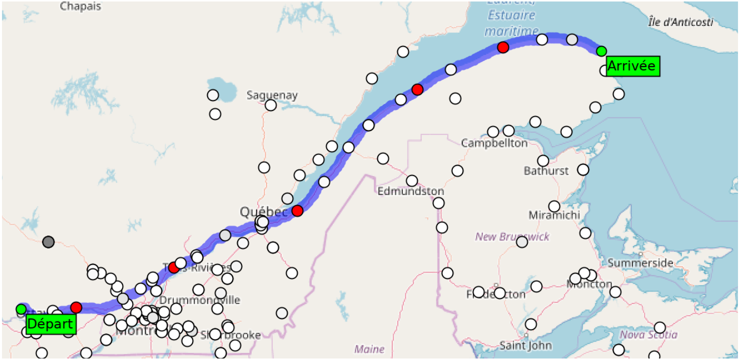
- Le nombre de bornes rapide pour VÉ croit moins vite que le nombre de VÉ.
- L'attente risque d'augmenter.
- Certaines régions sont mal desservies.
- Les longs trajets nécessitent une ou plusieurs recharges.

Objectif

Avoir un planificateur d'itinéraires pour VÉ qui :

1. tient compte des caractéristiques propres aux VÉ ;
2. considère les recharges à des bornes en milieu de trajet ;
3. considère le temps d'attente à chaque borne ;
4. minimise le temps de calcul.

Exemple visuel de la problématique



Formalisme – Problème

Réseau routier

Un réseau routier est représenté par un graphe orienté valué (V, E, λ) et par un ensemble S où :

- V est l'ensemble des positions considérées sur la carte ;
- E est l'ensemble des segments de routes ;
- $\lambda \colon E \to \mathbb{R}^+$ donne la longueur des arcs (ex : en km) ;
- S est l'ensemble des bornes de recharges (on suppose $S \subseteq V$).

Formalisme – Problème

Réseau routier

Un réseau routier est représenté par un graphe orienté valué (V, E, λ) et par un ensemble S où :

- V est l'ensemble des positions considérées sur la carte ;
- E est l'ensemble des segments de routes ;
- $\lambda \colon E \to \mathbb{R}^+$ donne la longueur des arcs (ex : en km) ;
- S est l'ensemble des bornes de recharges (on suppose $S \subseteq V$).

Planification pour VÉ

Un problème de planification pour VÉ (PPVÉ) est défini par le tuple $(M, \rho, \alpha, \omega)$ où :

- M est le réseau routier ;
- $\rho \in \mathbb{R}^+$ est l'autonomie du VÉ (en km) ;
- $\alpha, \omega \in V$ sont les nœuds de départ et d'arrivée.

Formalisme – Solution

Solution

Une **solution** à un PPVÉ $(M, \rho, \alpha, \omega)$ est un tuple (P, Q), où

- $P \subseteq V$ est la suite de nœuds à traverser dans la solution ;
- $Q \subseteq P$ contient les bornes où charger (et α, ω) ;
- $\forall i, d(Q_i, Q_{i+1}) \leq \rho$, où d est la distance dans le graphe.

Formalisme – Solution

Solution

Une **solution** à un PPVÉ $(M, \rho, \alpha, \omega)$ est un tuple (P, Q), où

- $P \subseteq V$ est la suite de nœuds à traverser dans la solution ;
- $Q \subseteq P$ contient les bornes où charger (et α, ω) ;
- $\forall i, d(Q_i, Q_{i+1}) \leq \rho$, où d est la distance dans le graphe.

Solution optimale

Une **solution optimale** est une solution (P, Q) qui minimise

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

où DT, CT et WT donnent le temps espéré de déplacement, de charge et d'attente.

Planificateur de base

Algorithme

Pour trouver le chemin le plus rapide de α à ω :

- Une matrice des distances entre chaque paire de bornes est précalculée ;
- La distance de α jusqu'à chaque borne, et de chaque borne jusqu'à ω est calculée (algorithme de Dijkstra 2 fois) ;
- Un graphe complet (V', E') est construit, où $V' = S \cup \{\alpha, \omega\}$.
- Les arcs plus longs que ρ sont retirés du graphe ;
- L'algorithme de A* de α à ω est lancé sur le nouveau graphe.

La complexité asymptotique temporelle totale est $\mathcal{O}(|V| \log |V| + |E|)$.

Éléments à considérer

- Recharge à des bornes en milieu de trajet

Éléments à considérer

- Recharge à des bornes en milieu de trajet
- Récupération d'énergie par le véhicule (topographie, freinage, etc.)

Éléments à considérer

- Recharge à des bornes en milieu de trajet
- Récupération d'énergie par le véhicule (topographie, freinage, etc.)
- Calcul rapide de l'itinéraire

Éléments à considérer

- Recharge à des bornes en milieu de trajet
- Récupération d'énergie par le véhicule (topographie, freinage, etc.)
- Calcul rapide de l'itinéraire
- Minimisation du temps d'attente

Energy A*

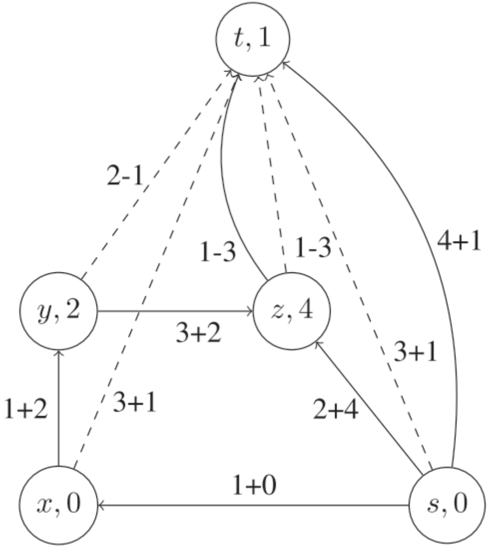
- Recharge à des bornes en milieu de trajet
- **Récupération d'énergie par le véhicule (topographie, freinage, etc.)** [5]
- Calcul rapide de l'itinéraire
- Minimisation du temps d'attente

5. Sachenbacher, M., M. Leucker, A. Artmeier, et J. Haselmayr (2011). Efficient Energy-Optimal Routing for Electric Vehicles. In Proceedings of the AAAI, pp. 1402–1407. AAAI Press.

Energy A* — Idée

- Considérer le poids des arêtes comme étant $c = c_L + c_P$;
- $C_L(e) = E_L(\ell(e), s(e))$, est la perte d'énergie dûe à l'environnement ;
- $C_P(a, b) = \pi(b) - \pi(a)$ est la différence d'énergie potentielle ;
- $\ell(e)$ est la longueur de l'arête ;
- $s(e)$ est la limite de vitesse du segment de route ;
- Il peut y avoir des arêtes négatives ;
- Pas de cycle négatif.

Energy A* — Illustration



Energy A* — Algorithme

- Utiliser l'algorithme de Johnson pour réétiqueter les arêtes ;
- Utiliser A* pour trouver le chemin optimal ;
- L'heuristique est donnée par $E_L(\|(a, b)\|, s_{\min})$;

Contraction de graphes

- Recharge à des bornes en milieu de trajet
- Récupération d'énergie par le véhicule (topographie, freinage, etc.)
- **Calcul rapide de l'itinéraire**[6]
- Minimisation du temps d'attente

6. Geisberger, R., Sanders, P., Schultes, D., Vetter, C. : Exact Routing in Large Road Networks Using Contraction Hierarchies. Transportation Science 46(3), 388–404 (2012)

Contraction de graphes — Idée

- Dans un graphe routier, la majorité des arêtes sont des routes secondaires ;
- La majeure partie de chaque trajet se fait sur des autoroutes ;
- On voudrait créer des **raccourcis** dans le graphe en ajoutant des arêtes.

Contraction de graphes — Construction des contractions

Définition

Soit un noeud N ayant des arêtes entrantes partant des noeuds $\{u_1, u_2, \ldots, u_k\}$ et ayant des arêtes sortantes allant aux noeuds $\{v_1, v_2, \ldots, v_p\}$. La **contraction** du noeud N se fait en :

- retirant N et ses arêtes incidentes ;
- ajoutant une arête entre chaque u_i et v_j tels que $< u_i, N, v_j >$ est un plus court chemin de u_i à v_j.

Contraction de graphes — Construction des contractions

Définition

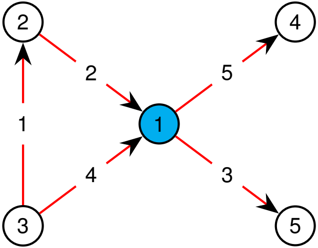
Soit un noeud N ayant des arêtes entrantes partant des noeuds $\{u_1, u_2, \ldots, u_k\}$ et ayant des arêtes sortantes allant aux noeuds $\{v_1, v_2, \ldots, v_p\}$. La **contraction** du noeud N se fait en :

- retirant N et ses arêtes incidentes ;
- ajoutant une arête entre chaque u_i et v_j tels que $< u_i, N, v_j >$ est un plus court chemin de u_i à v_j.

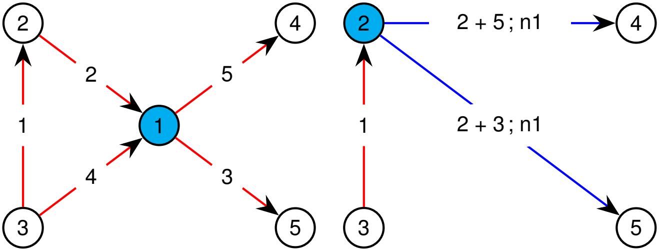
- Soit $S = \{s_1, s_2, \ldots, s_n\}$ un ordonnancement des noeuds du graphe ;
- On contracte les noeuds un par un dans cet ordre ;
- G^* est le graphe qui contient tous les noeuds et les arêtes initiales, ainsi que les raccourcis ajoutés par les contractions.

UQÀM

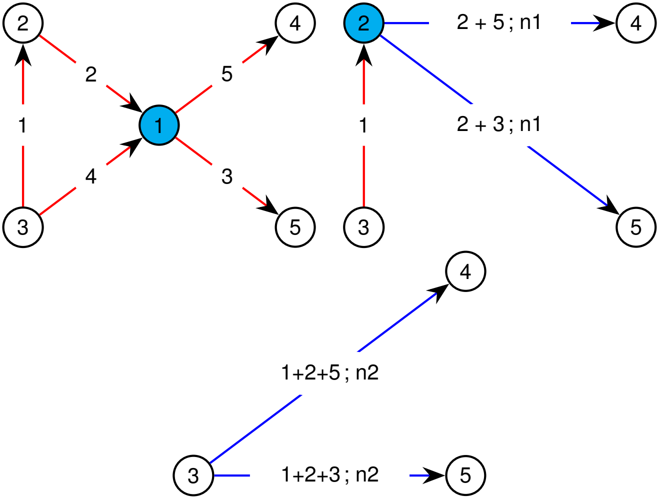
Contraction de graphes — Exemple 1/2



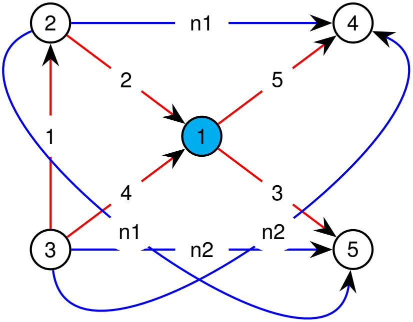
Contraction de graphes — Exemple 1/2



Contraction de graphes — Exemple 1/2



Contraction de graphes — Exemple 2/2



Contraction de graphes — Requête de s à t

- $G^*\uparrow = (S, \{(u, v) \in E \colon v > u\})$
- $G^*\downarrow = (S, \{(u, v) \in E \colon v < u\})$
- Faire un Dijkstra dans $G^*\uparrow$ à partir de s
- Faire un Dijkstra inversé dans $G^*\downarrow$ à partir de t
- Soit X l'ensemble des noeuds visités par les deux Dijkstras
- $d(s, t) = \min_{v \in X}(d(s, v) + d(v, t))$

Temps d'attente

- Recharge à des bornes en milieu de trajet
- Récupération d'énergie par le véhicule (topographie, freinage, etc.)
- Calcul rapide de l'itinéraire
- **Minimisation du temps d'attente**

Temps d'attente — Motivation

- Sans considération de l'occupation, on obtient (Départ, Borne1, Arrivée)
- E[temps d'attente B1] $= 0.8 * 15 = 12$ min
- E[temps d'attente B2] $= 0.1 * 15 = 1.5$ min
- On attend en moyenne $12 - 1.5 = 10.5$ min de plus à B1

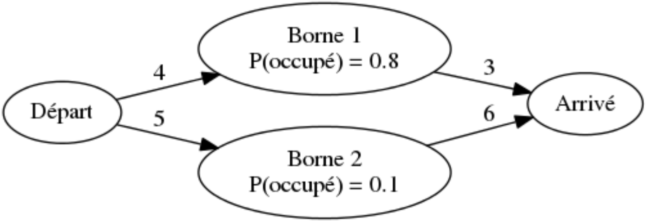


FIGURE – Problème lorsqu'on ne considère pas l'occupation des bornes

Temps d'attente — Technique 1

- Utilisation des données historiques de chaque borne *b*.
- Réétiquetage du graphe pour considérer ces données.

Données d'occupation à priori

Pour chaque borne *b*, la probabilité dynamique d'occupation est donnée par :

$$f_b \colon \{Lundi, \ldots, Dimanche\} \times \{0..23\} \to [0,1]$$
$$(d,h) \mapsto \mathbb{P}(b \text{ est occupée} \mid \text{Jour} = d \wedge \text{Heure} = h).$$

et le temps d'attente espéré lorsqu'occupée est donné par :

$$g_b \colon \{Lundi, \ldots, Dimanche\} \times \{0..23\} \to \mathbb{R}^+$$

UQÀM

Réétiquetage dynamique

Étiquetage considérant le temps d'attente

Soit $e = (u, v) \in E$. On définit l'étiquetage dynamique comme étant

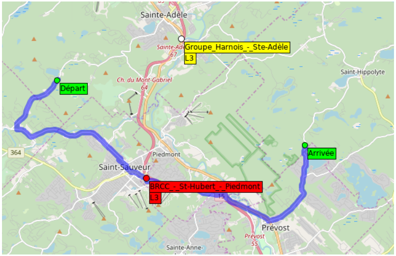
$$\xi \colon E \times \{Lundi, \dots, Dimanche\} \times \{0..23\} \to \mathbb{R}^+$$

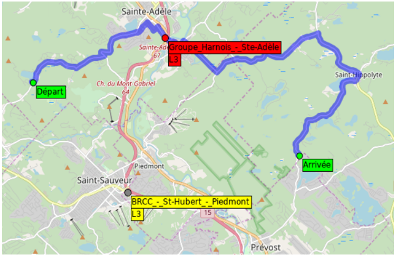
$$\xi(e, d, h) = \begin{cases} \lambda(e) & \text{si } u \notin S \\ \lambda(e) + f_u(d, h) \cdot g_u(d, h) \cdot \mu(e) & \text{si } u \in S \end{cases}$$

- Le poids des arcs dépend maintenant du temps d'arrivée ;
- on doit donc modifier l'algorithme de recherche de chemins (ex : 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

Temps d'attente — Effet sur les plans obtenus





Chemin retourné par le planificateur lundi midi vs mardi midi

Défauts de la technique précédente

- Elle n'utilise que des données connues à priori ;
- Le temps d'attente réel peut être bien pire que prévu ;
- Nous supposons qu'on a accès au temps d'attente en temps réel lorsqu'on conduit ;
- On peut alors réduire le temps de calcul en précalculant des plans alternatifs.

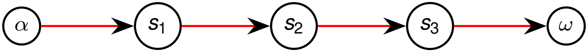
Deux cas extrêmes

1. Pas de chemins alternatifs ;
2. Une politique totale π : État $\rightarrow$ Action (ex : obtenue avec un MDP)

Nous voulons un compromis entre ces deux cas.

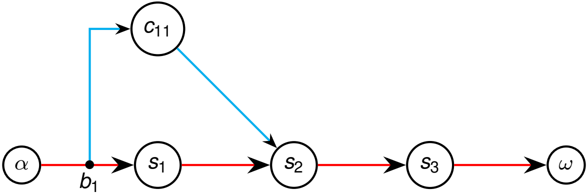
Temps d'attente — Technique 2

Générer un chemin alternatif pour chaque borne sur le chemin initial.



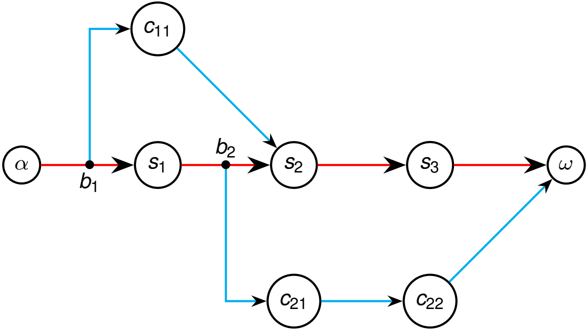
Temps d'attente — Technique 2

Générer un chemin alternatif pour chaque borne sur le chemin initial.



Temps d'attente — Technique 2

Générer un chemin alternatif pour chaque borne sur le chemin initial.



Temps d'attente — Technique 2

Algorithme 1 Génération de chemins alternatifs

1: chemin de base $\leftarrow$ planificateur de base (incluant technique 1)
2: **pour toute** $s_i \in Q$ **faire**
3: supposer temporairement que $f_{s_i} \equiv 1$ ▷ Simule que s_i est occupée
4: nouveau chemin $\leftarrow$ planificateur de base (incluant technique 1)
5: **si** nouveau chemin $=$ ancien chemin **alors**
6: **continue** ▷ Pas d'alternative pour cette borne
7: $b_i \leftarrow$ dernier nœud en commun du préfixe des deux chemins
8: mettre le nouveau chemin comme alternative sur le nœud b_i

L'algorithme nous retourne la politique partielle

$$\pi : V \rightarrow V^2$$

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

UQÀM

Utilisation de la politique

Algorithme 2 Exécution du plan

procédure EXECUTER_PLAN(π)
    $n \leftarrow \alpha$
    tant que $n \neq \omega$ **faire**
        $(x, y) \leftarrow \pi(n)$
        si $y = - \vee \neg\text{occupée}(x)$ **alors**
            $n \leftarrow x$
        sinon
            $n \leftarrow y$
        déplacer le VÉ jusqu'au nœud n

Méthodologie de test

- Les données réelles de la carte proviennent du projet OpenStreetMap.
- Les données des bornes (position et données historique d'occupation) proviennent du Circuit Électrique.
- Le territoire utilisé est la province du Québec :
  - 2 923 013 nœuds ;
  - 5 907 672 arcs ;
  - 1318 bornes de recharge (1178 L2 et 140 L3).
- 1000 requêtes générées :
  - Autonomie variant entre 100 et 550 km ;
  - α et ω sont choisis au hasard parmi tous les nœuds ;
  - Longueur des itinéraires entre 200 et 1500 km

Temps d'attente — Résultats

Paramètres		Base		Réétiquetage				Chemins alternatifs			
RB	MP	A min	AD min	A min	AD min	RA %	RT min	A min	AD min	RA %	RT min
R140	× 1	10.2	350.1	3.3	343.9	-67.5	-6.2	2.2	342.8	-78.8	-7.2
R140	× 2	22.7	362.5	6.0	347.0	-73.6	-15.5	4.1	345.3	-82.0	-17.2
R140	× 3	37.5	377.3	7.7	349.0	-79.3	-28.3	7.2	348.5	-80.9	-28.7
R140	Aléa	51.3	391.2	10.9	352.6	-78.7	-38.5	9.8	351.6	-80.9	-39.5
R1318	× 1	19.4	356.0	2.2	339.2	-88.6	-16.7	1.7	338.7	-91.5	-17.3
R1318	× 2	37.5	374.0	4.0	341.2	-89.3	-32.8	3.0	340.3	-92.0	-33.7
R1318	× 3	50.5	387.1	6.5	343.9	-87.1	-43.1	4.7	342.3	-90.6	-44.8
R1318	Aléa	62.0	398.6	6.9	345.3	-88.8	-53.3	6.0	344.4	-90.4	-54.2
A250	Aléa	29.3	368.2	12.0	354.2	-59.0	-13.9	10.3	353.1	-64.8	-15.1
A500	Aléa	28.9	363.4	9.4	346.9	-67.6	-16.5	8.3	346.1	-71.2	-17.3
A1000	Aléa	28.7	362.5	7.4	343.8	-74.2	-18.7	6.5	343.1	-77.3	-19.4
A2000	Aléa	27.1	359.9	4.9	340.0	-81.9	-19.9	3.8	339.0	-86.0	-20.9

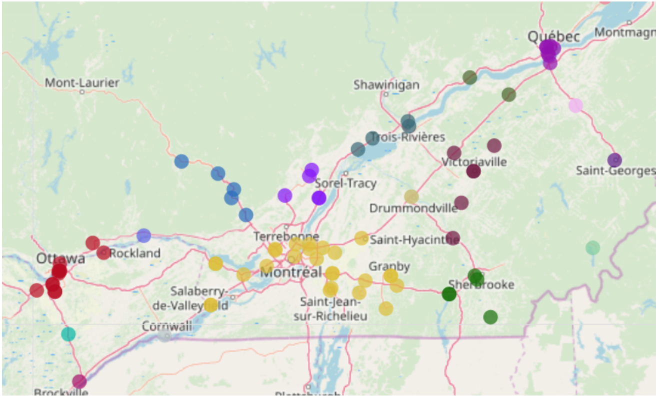
RB : Réseau de bornes (où R = réelles et A = artificielles) ;
MP : Modificateur des probabilités ; **A** : Temps d'attente moyen ; **AD** : Temps total moyen (Attente + Déplacement) ;
RA : Réduction de l'attente (vs Base) ; **RT** : Réduction du total (vs Base).

Regroupement de bornes

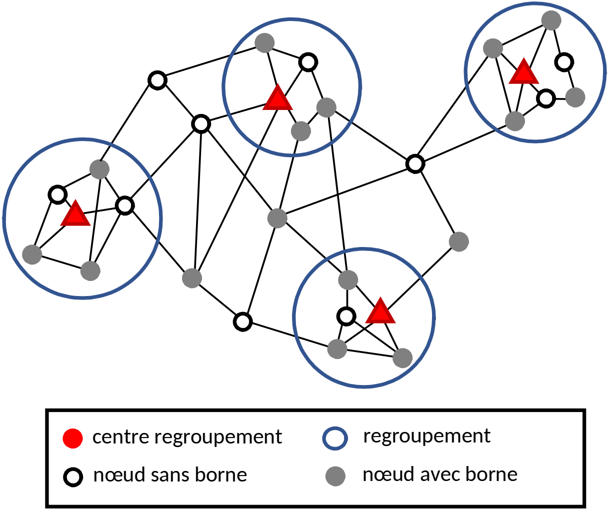
- Recharge à des bornes en milieu de trajet
- Récupération d'énergie par le véhicule (topographie, freinage, etc.)
- **Calcul rapide de l'itinéraire**
- Minimisation du temps d'attente

Regroupement de bornes — Idée

- Il y a 140 bornes rapides (L3) et 1178 bornes L2 ;
- La plupart d'entres elles sont situées dans les mêmes secteurs (grandes villes, sites touristiques, etc.) ;
- Regrouper les bornes rapprochées diminuerait grandement la taille du graphe.



Regroupement de bornes — Idée



Regroupement de bornes — Algorithme

$d_{\max}$: distance maximale entre une borne et le nœud central de son regroupement

Algorithme 3 Génération des regroupements de bornes de recharge

1: Trouver les bornes $S_1, S_2 \in S$ les plus rapprochées
2: **tant que** distance $(S_1, S_2) \leq d_{\max}$ **faire**
3: Trouver le nœud m à mi-chemin de S_1 et S_2
4: Trouver $C = \{s \in S | dist(s, m) \leq d_{\max}\}$ ▷ $S_1, S_2 \in C$
5: $S \leftarrow (S \setminus C) \cup \{m\}$
6: Trouver les bornes $S_1, S_2 \in S$ les plus rapprochées

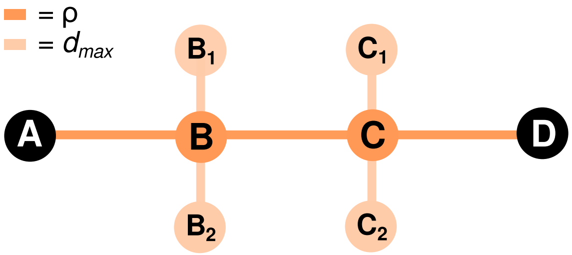
La complexité temporelle du regroupement est $\mathcal{O}(K(|S|^2 + |V|))$
où K est le nombre de clusters générés.

Regroupement de bornes — Modification à l'autonomie

Ajustement de l'autonomie considéré

Pour que le chemin retourné par le planificateur soit réalisable, l'autonomie considérée doit être :

- $\rho' = \rho - d_{\text{max}}$ pour le premier et dernier regroupement ;
- $\rho' = \rho - 2d_{\text{max}}$ entre les regroupements.

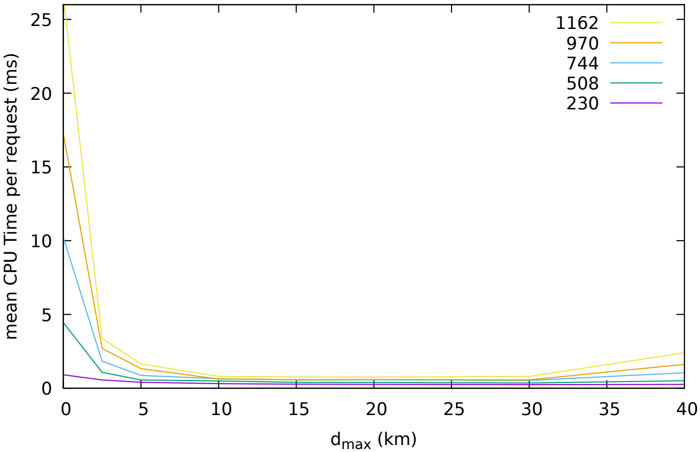


Regroupement de bornes — Résultats

Paramètres		Regroupements		Version base		Version amortie	
Bornes	d_{max}	Regroup.	JDIR	FR	CT	FR	CT
#	km	#	%	%	ms	%	ms
1162	0.0	1162	0.0	0.0	26.563	0	26.563
1162	2.5	487	0.0	0.0	3.385	0	3.385
1162	5.0	342	0.2	0.4	1.541	0	1.647
1162	10.0	236	0.2	0.8	0.588	0	0.801
1162	15.0	188	0.6	0.9	0.523	0	0.762
1162	20.0	150	1.0	1.4	0.382	0	0.754
1162	30.0	111	2.3	2.0	0.265	0	0.796
1162	40.0	87	2.8	8.2	0.226	0	2.404

JDIR : Journey duration increase rate ; **FR** : Failure rate ; **CT** : Computation time

Regroupement de bornes — Résultats

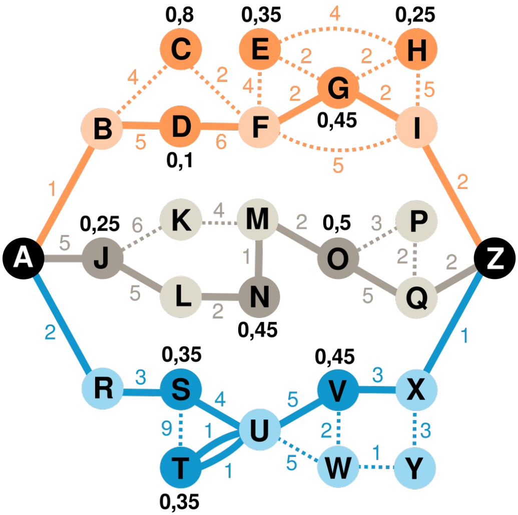


Algorithme global

Algorithme 4 Planification complète pour VÉ

```
Calculer la matrice D et le chemin le plus court reliant chaque paire de bornes
Construire le s-graphe contenant chaque borne de recharge
pour toute requête (ρ, α, ω) faire                    ▷ Tant qu'il y a des requêtes à traiter
    Exécuter Dijkstra partant de α sur le graphe original
    Exécuter Dijkstra partant de ω sur le graphe transposé
    Ajouter α et ω au s-graphe et ajouter les arcs de longueur ≤ ρ
    Exécuter l'algorithme A* de α à ω sur le s-graphe pour trouver la suite Q
    si le chemin est impossible alors                 ▷ Stratégie amortie
        Exécuter A* sur le s-graphe (sans regroupements) pour trouver Q
    Trouver la suite P à l'aide de Q en utilisant les chemins déjà calculés
    Retirer α et ω du s-graphe et réajuster les arcs
    Envoyer la solution (P, Q)
```

Exemple 1/3 : Réseau routier



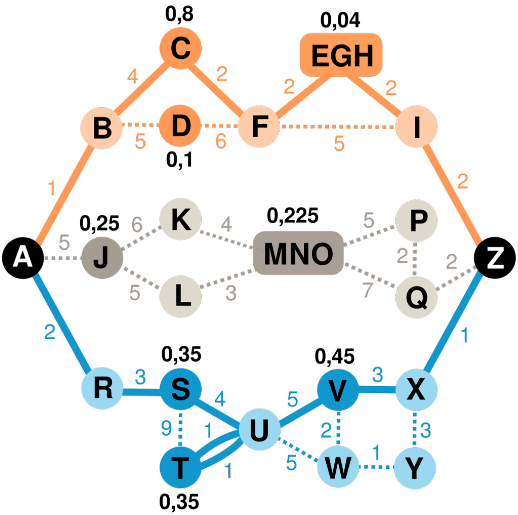
$$\text{Range} = 8$$

$$d_{max} = 2$$

$$\alpha = \text{A}$$

$$\omega = \text{Z}$$

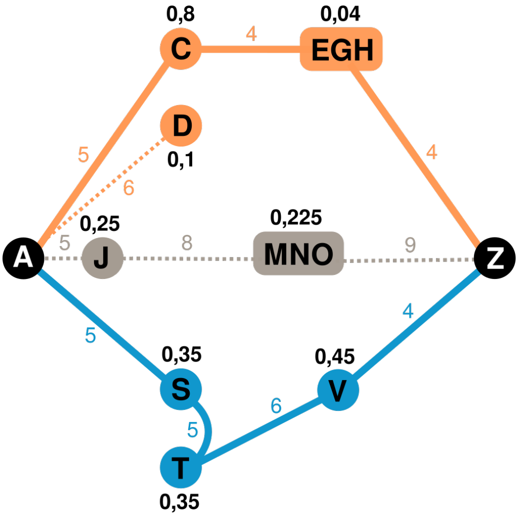
Exemple 2/3 : Regroupement de bornes



$$\text{Range} = 8$$
$$d_{max} = 2$$
$$\alpha = \text{A}$$
$$\omega = \text{Z}$$

UQÀM

Exemple 3/3 : Construction du graphe simplifié



$$\text{Range} = 8$$
$$d_{max} = 2$$
$$\alpha = \text{A}$$
$$\omega = \text{Z}$$

UQÀM

Conclusion

- Le regroupement de bornes de recharge permet de réduire d'un facteur 35 le temps de calcul ;
- Les techniques pour la considération de l'attente ont divisé par 4 le temps d'attente (17.3 minutes de moins en moyenne).

Reconnaissance

CRSNG NSERC

Fonds de recherche sur la nature et les technologies Québec

Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) et le Fonds de recherche du Québec — Nature et technologies (FRQNT).

UQÀM