Highway Hierarchies : Recherche de plus court chemin

L'algorithme de recherche du plus court chemin à source unique le plus connu est celui proposé par Dijkstra en 1959. Cependant, lorsque la taille du graphe devient trop importante, le temps d'exécution devient prohibitif. Par exemple, pour représenter le réseau routier européen, la compagnie PT AVG fournit un graphe contenant 20 000 000 nœuds (S) et autant de chemins (A). Implémenté avec un tas min, l'algorithme de Dijkstra donnerait une complexité de O((A + S)log S), soit un temps d'exécution d'environ 10s sur un processeur à 1GHz. De même, on ne peut précalculer et stocker tous les plus courts chemins : il faudrait 1000 To pour stocker seulement les distances. Des heuristiques peuvent alors permettre d'accélérer l'exécution de l'algorithme de Dijkstra. Cependant, ils ne fournissent pas nécessairement le chemin le plus court, mais seulement une approximation.

On s'intéresse ici à l'algorithme de hiérarchies d'autoroutes (Highway Hierarchies) qui, grâce à un prétraitement des données, permet d'obtenir rapidement un plus court chemin exact.

Table des matières

1 Principe

Cet algorithme repose sur une idée intuitive : plus on s'éloigne du point de départ, plus on utilise des routes de « haut niveau », des départementales puis des nationales, puis des autoroutes, et on n'utilise à nouveau des routes de « bas niveau » seulement une fois proche de l'arrivée. L'algorithme va donc créer une hiérarchie d'autoroutes puis appliquer dessus l'algorithme de Dijkstra assorti de quelques restrictions permettant de le rendre plus rapide.

2 Prétraitement des données

Le graphe fourni $G_0 = (S_0, A_0)$ doit être non orienté à poids strictement positifs, et on fixe un paramètre H appartenant à $\mathbb{N}^*$. Pour $s \in V$, l'exécution de l'algorithme de Dijkstra à partir de s permet de définir le rang $r_s(v)$ d'un nœud $v \in S_0$: $r_s(s) = 0$, et si v_1 est premier nœud établi par l'algorithme de Dijkstra après s, $r_s(v_1) = 1$... On définit alors $d_H(s) = d(s, v)$ (longueur du plus court chemin de s à v) où $r_s(v) = H$; puis $\mathcal{N}(s) = \{v \in S_0 | d(s, v) \leq d_H(s)\}$. $\mathcal{N}(s)$ est appelé le voisinage de s. L'algorithme va alors construire le réseau d'autoroutes $G_1 = (S_1, A_1)$.

2.1 Définition

Soit une arrête $(u, v) \in A_0$. (u, v) appartient à A_1 si et seulement si :

$$\text{P=<s, ..., u, v, ..., t> est un plus court chemin avec v} \notin \mathcal{N}(s) \text{ et u} \notin \mathcal{N}(t) \tag{1}$$

S_1 est le plus grand sous ensemble de S_0 tel que G_1 ne contienne pas de nœuds isolés.

2.2 Construction

Initialisation $E_1 = \varnothing$

Pour chaque $s_0 \in A_0$: On construit B l'arbre de plus court chemin partiel d'origine s_0. Afin de limiter la taille de B, on attribue aux nœuds atteints par l'algorithme de Dijkstra l'état actif ou passif :

- s_0 est active ;
- un nœud atteint prend l'état d'activation de son parent ;
- si un nœud v est établi par le plus court chemin $P =< s_0, s_1, ..., v >$ et si $|\mathcal{N}(s_1) \cap \mathcal{N}(v)| \leq 1$ alors v devient passif.

Lorsqu'il n'y a plus aucun nœud actif non établi, la croissance de B est arrêtée. Pour chaque arrête (u, v) de B, (u, v) est ajoutée à E_1 si (*) est vérifiée avec $s = s_0$ et t une feuille de B.

2.3 Mise en place de la hiérarchie

On construit de même G_2 à partir de G_1, puis G_3 à partir de G_2... jusqu'à G_L. On relie ensuite ces différents niveaux :

— Chaque nœud $v \in S_0$ donne $v_i \in S_i$ si $v \in S_i$.
— On appelle arrête horizontale les élément de A_i.
— On ajoute des arrêtes verticales : orientées, de poids 0 reliant v_i *à* v_{i+1}.

3 Requêtes

On utilise l'algorithme de Dijkstra auquel on ajoute la restriction suivante : dans chaque niveau $l \leq L$, on ne relâche pas d'arrêtes horizontales sortant du voisinage au niveau l de $v*$, où $v*$ est le nœud par lequel le niveau l a été atteint sur le chemin menant à v, nœud en cours de traitement.

Pour obtenir le plus court chemin de s à t, on exécute cet algorithme depuis s_0, nœud correspondant à s dans G_0, et depuis t_0. Une fois ces deux recherches terminées, on teste chacun des nœuds établis par les deux recherches afin de trouver le plus court chemin.

4 Implémentation

4.1 Données

J'ai implémenté la création de la hiérarchie d'autoroutes sur 500 000 km de routes du réseau classé (autoroutes, nationales, départementales) français fournis par l'IGN [1]. Le graphe original fournit pour chaque arrête sa longueur et sa catégorie. J'ai assigné une vitesse moyenne à chacune de ces catégories : autoroutes (130 km/h), liaisons principales (90 km/h), laisons régionales (70 km/h) et liaisons locales (50 km/h), et utilisé le temps de parcours ainsi calclulé comme pondération. Le graphe est composé de 204 477 nœuds et 298 275 arrêtes.

4.2 Résultats obtenus pour H = 125

Afin de présenter les résultats les plus lisibles possibles, j'ai choisi de ne garder que ceux correspondant à H = 125. Les autres valeurs de H correspondaient à un trop grand nombre de niveaux.

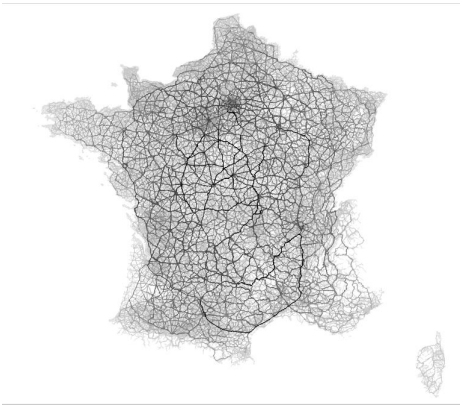

FIGURE 1 – Les 7 niveaux fournis par l'algorithme pour H = 125 : plus l'arrête est de niveau élevé plus elle est foncée

Lorsque le niveau augmente, le maillage des routes de ce niveau est de moins en moins dense, mais couvre néanmoins presuqe toute la superficie de la France.

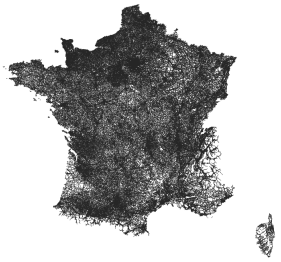

FIGURE 2 – Niveau 0

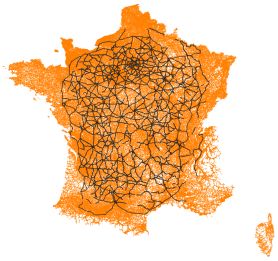

FIGURE 4 – Niveau 5

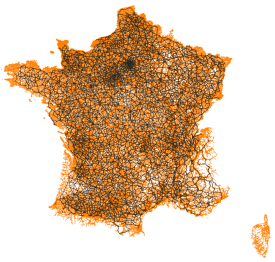

FIGURE 3 – Niveau 3

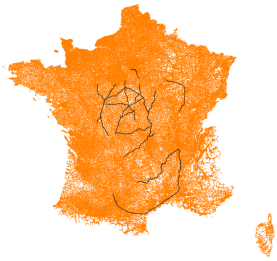

FIGURE 5 – Niveau 7

On superpose ici les autoroutes réelles (en orange) et les routes de niveau 5 obtenues avec l'algorithme. On remarque que la définition algorithmique de ce qu'est une autoroute donne des résultats proches de la hiérarchie routière réelle.

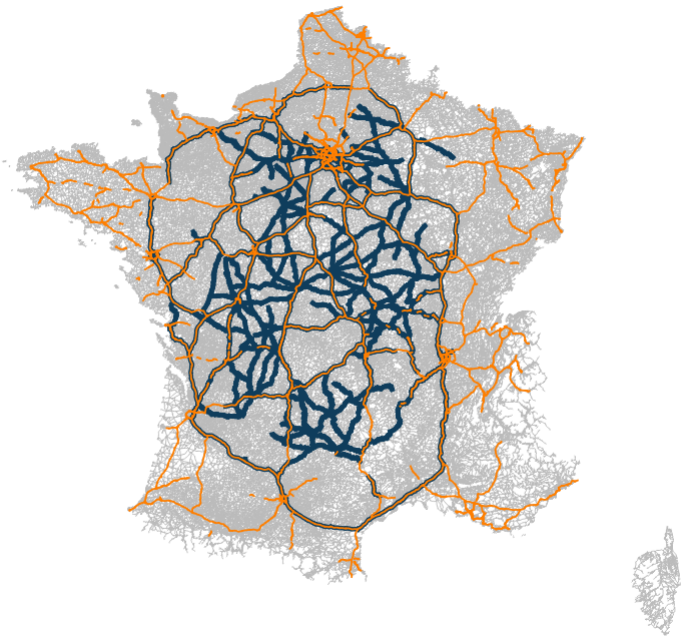

FIGURE 6 – Superposition des autoroutes réelles et du niveau 5 pour H = 125

4.3 Performances

Précalcul

H	Temps d'exécution[h :min]
50	1 :07
75	2 :05
125	5 :01

TABLE 1 – Temps de calcul de la hiérarchie par mon implémentation sur un processeur 2.3Ghz

Les temps de précalcul sont courts, et d'autres optimisations seraient possibles.

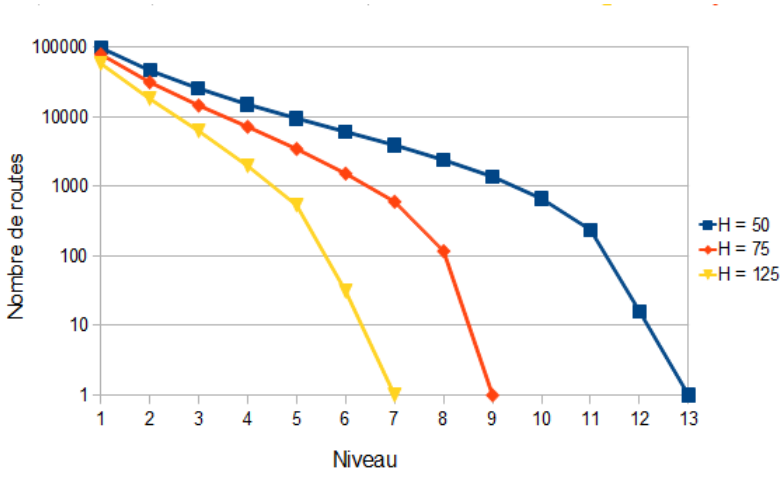

FIGURE 7 – Taille des réseaux d'autoroutes

On remarque que le nombre de niveaux est faible même pour des valeurs de H peu importantes. La quantité de données à stocker est donc de l'ordre de la taille du graphe fourni.

Requêtes Sur le graphe fourni par la compagnie PTV AVG représentant le réseau routier européen (18 029 721 nœuds et 22 217 686 arêtes), l'implémentation proposée par P. Sanders et D. Schultes[2] fournit un résultat 2654 fois plus rapidement que l'algorithme de Dijkstra (accélération moyenne pour 10 000 requêtes où source et objectif sont choisis aléatoirement).

Références

[1] IGN. Route 500, 2012. http ://professionnels.ign.fr/route500.

[2] Dominik Schultes Peter Sanders. *Highway Hierarchies Hasten Exact Shortest Path Queries.* Universitat Karlsruhe (TH), 76128 Karlsruhe, Germany, and Universitat des Saarlandes, 2005.