26609 PUPAT Martin

Congestion en milieu urbain : fluidifier le trafic grâce au routage dynamique

Pour me conduire à mon lycée, mes parents utilisaient une application de navigation pour éviter les embouteillages. Les autres usagers aussi... Et me voilà coincé et encore en retard en cours. Comment l'application aurait-elle pu mieux anticiper cela ? Pour résoudre ce problème, j'ai décidé d'en faire mon sujet de TIPE.

La route est une infrastructure intégrante de la ville, et nombreux sont les citadins qui l' empruntent de manière quotidienne, à l'aide d'outils de navigation. Les communes œuvrant aussi au mieux pour garantir calme et fluidité routière aux habitants, le sujet que j'étudie jongle avec plusieurs aspects de la ville.

Positionnement thématique (ÉTAPE 1) :

- INFORMATIQUE (Informatique Théorique)
- INFORMATIQUE (Informatique pratique)
- MATHEMATIQUES (Autres)

Mots-clés (ÉTAPE 1) :

Mots-clés (en français)	**Mots-clés (en anglais)**
Routage dynamique	*Dynamic routing*
Algorithme de plus court chemin	*Shortest path algorithm*
Coordination décentralisée de véhicules	*Decentralized vehicle coordination*
Complexité algorithmique	*Algorithmic complexity*
Congestion routière	*Road congestion*

Bibliographie commentée

La démocratisation des moyens de transport personnels, l'augmentation des distances et durées des trajets quotidiens, et la saturation des infrastructures urbaines a montré ces dernières décennies la nécessité pour les voyageurs pendulaires de connaître les chemins les plus rapides possibles pour se rendre à leur lieu de travail, tout en saturant au minimum les routes pour le confort des résidents urbains.

Une première approche « naïve » pour optimiser les temps de trajet est celle du routage statique : il s'agit essentiellement, étant donné un état figé d'un réseau routier, de trouver algorithmiquement le chemin le plus rapide entre deux points. Il existe déjà des solutions à ce problème [1] : l'algorithme de Dijkstra bi-directionnel produit un chemin optimal en complexité semi-linéaire ; les algorithmes A* et de Dijkstra avec périmètre limite de recherche produisent un chemin quasi-optimal en complexité linéaire.

La modélisation avec un niveau de complexité supérieur est celle dite du « commerçant

dynamique » [2] : on se place dans le cas d'un réseau routier statique, mais où le trajet du véhicule n'est pas défini, les destinations étant en effet générées aléatoirement (demandes de rendez-vous des clients dans la ville) et le but est d'une part de minimiser le temps d'attente des clients et d'autre part de maximiser le nombre de clients visités. Bien que l'approche naïve du « premier arrivé, premier servi » se montre peu efficace [2, 3], en utilisant diverses heuristiques (division de la ville et traitement zone par zone, utilisation de plusieurs véhicules, visite favorisée des voisins proches), des temps d'attente moyens de la part des clients environ proportionnels avec leur nombre sont envisageables [3].
Malheureusement, cette approche ne prend pas en considération des problèmes tels que les feux rouges ou les accidents en chemin. Il faut investiguer du côté du routage dynamique, où le réseau est susceptible de changer en temps réel, et recalculer le chemin optimal du véhicule depuis sa position actuelle chaque fois qu'un choix doit être fait. Pour des véhicules individuels, la réutilisation des algorithmes A* et de Dijkstra avec périmètre limite en recalculant uniquement les sommets du graphe routier (les intersections) ayant changé (soit eux-mêmes, soit une de leurs arêtes / routes) s'avèrent produire des chemins quasiment optimaux [4]. Leur complexité de recalcul est en pratique linéaire avec le nombre d'arêtes modifiées [5], mais peut être encore améliorée grâce à l'algorithme BallString [5], d'une complexité logarithmique en pratique, mais donnant en contrepartie des chemins moins optimaux que les précédents.
Ainsi, il existe une solution satisfaisante au problème du routage dynamique pour un véhicule personnel individuel. Mais qu'en est-il d'un nombre important de véhicules ? Leur impact sur la congestion globale n'est plus négligeable ; il faut donc prendre cela en compte. Deux méthodes sont possibles [6]. On peut calculer les trajets de tous les véhicules sur un graphe routier vide, puis refaire cette opération en tenant compte de la congestion anticipée par les simulations précédentes (approche « centralisée »). On peut aussi calculer une première fois tous les chemins, puis recalculer sur le vif le chemin à prendre du véhicule individuel en tenant compte de la congestion actuelle à des moments-clefs, lorsqu'il doit prendre une décision par exemple (approche « décentralisée »).
En se servant de l'algorithme A* (largement utilisé précédemment [1, 3, 4, 5]), l'approche décentralisée permet le passage de plus de véhicules indépendamment de la saturation du réseau, et donc une meilleure congestion. Elle reste valide à de plus grandes échelles, par le biais d'une représentation macroscopique des flux de véhicules et non des véhicules individuels. Toutefois, aucune étude de complexité n'a été faite, dans chaque cas, ni d'étude pratique du problème pour essayer de trouver le meilleur compromis congestion / temps de trajet possible, ce que je me propose donc de faire dans ce TIPE.

Problématique retenue

Comment conduire à leur destination le plus rapidement possible un grand nombre de véhicules dans un environnement urbain en constante évolution, tout en minimisant les coûts de calcul et la saturation des infrastructures ?

Objectifs du TIPE du candidat

1. Implémenter l'algorithme de routage dynamique et décentralisé commenté dans la bibliographie en OCaml.
2. Décider d'un modèle mathématique adapté pour représenter les demandes des véhicules en fonction du temps.
3. Faire une analyse de la complexité de cet algorithme ; étudier la saturation des routes et les temps de trajets engendrés. Comparer ces trois points à des algorithmes de routage statique.
4. Explorer des possibilités d'amélioration en complexité de l'algorithme tout en gardant des résultats intéressants pour les utilisateurs sur des graphes routiers urbains réels.

Références bibliographiques (ÉTAPE 1)

[1] DEPARTMENT OF ENGINEERING SCIENCE : Fast Shortest Path Algorithms for Large Road Networks *: orsnz.org.nz/conf36/papers/Engineer.pdf*

[2] DIMITRIS J. BERTSIMAS, GARRETT VAN RYZIN : Stochastic and Dynamic Vehicle Routing with General Demand and Interarrival Time Distributions *: doi.org/10.2307/1427801*

[3] DAVID LARSEN : The Dynamic Vehicle Routing Problem *: ISSN 0909-3192*

[4] EDWARD P.F. CHAN, YAYA YANG : Shortest Path Tree Computation in Dynamic Graphs *: doi.org/10.1109/TC.2008.198*

[5] GIACOMO NANNICINI, LEO LIBERTI : Shortest Paths on Dynamic Graphs *: doi.org/10.1111/j.1475-3995.2008.00649.x*

[6] MELANIE SMITH, ROGER MAILLER : The Effect of Congestion Frequency and Saturation on Coordinated Traffic Routing *: doi.org/10.1007/978-3-642-25044-6*

DOT

[1] *: Septembre - novembre 2022 : J'ai fait des recherches bibliographiques dans le thème du routage de voitures. Cela m'a fait découvrir le problème du routage dynamique, que j'ai généralisé à plusieurs agents comme sujet de TIPE.*

[2] *: Novembre 2022 - janvier 2023 : En faisant des recherches peu fructueuses sur mon problème en raison de la nouveauté du domaine, j'ai trouvé l'existence de l'algorithme A* itéré, et je me suis inspiré de mes recherches précédentes pour créer l'algorithme BallString simplifié.*

[3] *: Décembre 2022 - avril 2023 : J'ai initialement commencé à implémenter ces algorithmes en Caml, mais puisque j'utilisais une bibliothèque Python pour lire les données servant à la création de mon graphe, j'ai changé de langage entre temps et opté pour ce dernier.*

[4] *: Mars - Mai 2023 : J'ai étudié de manière théorique la complexité des deux algorithmes, mais j'ai utilisé un résultat de ma bibliographie pour le second faute de temps. Cela m'a permis d'avoir une base pour étudier mes résultats expérimentaux.*

[5] *: Avril 2023 : J'ai généré les courbes de données en exécutant mon programme en modifiant plusieurs paramètres, et en le peaufinant au passage pour simuler des conditions plus réalistes.*

[6] *: Mai 2023 : L'analyse des données créées dans les deux jalons précédents m'a permis d'interpréter mes résultats et de tirer des conclusions au vu de mon questionnement initial.*

[7] *: Mai - Juin 2023 : Grâce à mon travail au long de l'année, j'ai construit ma présentation, en traçant des diagrammes et des courbes pour faciliter la compréhension du sujet, et je me suis entraîné sur ma présentation.*