Implant stratégique d’une école avec un diagramme de Voronoï

L’accès à la scolarité est un problème auquel chaque ville doit se confronter, en implanter une s’inscrit dans les nouvelles infrastructures majeures pour celles-ci.

L’aspect très pratique du problème combiné à l’algorithmie utilisée pour le résoudre me semble être au plus proche du métier que je veux pourvoir exercer à l’avenir.

Positionnement thématique

INFORMATIQUE (Informatique théorique), INFORMATIQUE(Informatique pratique).

Mots-Clés (français)	**Mots-Clés** (anglais)
Diagramme de Voronoï	Voronoi Diagram
Ecole	School
Structure de donnée Winged-Edge	Winged-Edge Data Structure
Complexité moyenne	Complexity in average
Proximité	Proximity

Bibliographie commentée

Un diagramme de Voronoï est pavage du plan en cellules à partir d’un ensemble discret de points appelés « germes ». Chaque cellule enferme un seul germe, et forme l'ensemble des points du plan plus proches de ce germe que d'aucun autre.[1]

La structure de données Winged-Edge est un moyen de représenter des maillages polygonaux dans la mémoire de l'ordinateur. C'est un type de représentation qui décrit explicitement la géométrie et la topologie des faces, des arêtes et des sommets lorsque trois surfaces ou plus se rejoignent et se rencontrent sur une arête commune.[1][2]

Un premier algorithme pour implémenter un diagramme de Voronoi serait l’algorithme de Shamos et Hoey qui théoriquement s’exécute en O(nlog(n)), soit la meilleure complexité atteignable en pire cas, s’effectuant à partir de la méthode diviser pour régner [1][4]

Un autre algorithme serait une méthode incrémentale qui construit le diagramme de Voronoï en ajoutant les points un par un et en adaptant de diagramme au fur et à mesure. Elle consiste à regarder dans quelle cellule figure notre nouveau point, à tracer ensuite la bissectrice entre notre point et celui à qui appartient cette cellule, jusqu'à intersecter les bords de la cellule. On poursuit en traçant la bissectrice entre notre point et celui de la cellule adjacente à la première arête intersectée et on répète le procédé jusqu'à avoir tracé la cellule de notre nouveau point, retirant les arêtes intérieures à cette cellule.[1]

Cet algorithme s'exécute en $O(n^2)$ mais en pratique, il est possible de le réduire à une complexité en $O(n)$. Un premier moyen de raffiner la complexité est de rendre plus « intelligent » notre postulat de départ lorsque l'on ajoute un point. En effet, à chaque ajout de point, il nous est obligatoire de parcourir les points jusqu'à trouver le plus proche (celui qui possède la cellule dans laquelle le nouveau point se trouve), un algorithme de recherche de plus proche voisin va nous permette de vite trouver ce plus proche point.[1] Un second moyen va être de réduire au mieux la quantité d'arêtes, de points ou autres éléments de structure à supprimer une fois un nouveau point ajouté, et ceci en essayant de réarranger l'ordre de nos points pour s'approcher de la répartition la plus uniforme possible, cela facilitera aussi les choix de l'algorithme précédent. La technique utilisée est alors la « bucketing technique » qui consiste à organiser la structure qui décompose l'espace, à partir de laquelle les données spatiales sont extraites, en régions appelées compartiments. Cette méthode sera ici employée à l'aide d'un arbre enraciné quaternaire pour diviser les données.[1][3]

Problématique

Dans quelle mesure l'emploi d'un diagramme de Voronoï permet de déterminer la position idéale d'implantation d'une nouvelle école ?

Objectifs

Dans le but d'obtenir une division parcellaire de la ville et en déduire les zones trop éloignées des centres d'éducation, l'objectif est d'implanter la méthode incrémentale pour le diagramme de Voronoï puisqu'elle est de nature plus simple comparée aux autres algorithmes et permet la meilleure complexité moyenne.

Références bibliographiques

[1] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, Sung Nok Chiu : Spatial Tessellations : Concepts and Applications of Voronoï Diagrams : *Seconde Edition, Chapitre 4.*

[2] Pr. Donald H. House : Winged-Edge Data Structure : *https://people.computing.clemson.edu/~dhouse/courses/405/papers/winged-edge.pdf*

[3] Takao OHYA, Masao IRI, Kazuo MUROTA : A fast Voronoi Diagram algorithm with quaternary tree bucketing.

[4] Michael Ian Shamos, Dan Hoey : Closest-points Problems.