

TIPE

Comment optimiser les tracés des routes dans une ville en expansion ?

I. Objectif :

Ce TIPE s'inscrit dans un projet en binôme, constitué de deux parties bien distinctes. Dans une première partie, l'autre membre de mon binôme étudie la croissance de villes sur plusieurs décennies grâce à des photos satellites. Il cherche ainsi à prédire, à partir de la croissance observée, la croissance future de la ville étudiée.

À partir de son travail, je dispose d'une surface de « future ville » prédite. Mon objectif est de déterminer une répartition optimale des routes pour cette surface, et ainsi de proposer à la ville un plan de tracé des routes adapté à sa croissance. Par « optimal », j'entends ici que la longueur moyenne d'un trajet d'un point à un autre de la ville est la plus courte possible, tout en s'assurant que la somme des longueurs des routes de la ville reste acceptable. On s'assure ainsi que l'on n'a pas réduit la longueur moyenne des plus courts chemins au prix de la construction d'un nombre déraisonnable de routes. L'optimalité sera mesurée par une fonction de coût qui prendra en compte ces deux éléments.

J'ai choisi d'utiliser la méthode du recuit simulé.

II. Le recuit simulé en quelques mots : (la version que j'utilise)

L'algorithme prend en entrée un état initial de l'objet étudié (ici une ville), il dispose d'une fonction de coût et il cherche un état de l'objet pour lequel la fonction de coût prend une valeur minimale.

Pour cela l'algorithme répète un très grand nombre de fois l'opération suivante :

Il calcule un état aléatoire de l'objet proche de son état au début de l'opération, et regarde si la fonction de coût prend une valeur plus grande ou plus petite pour ce nouvel état que pour le précédent.

Si la valeur est plus petite, le nouvel état est accepté et devient l'état actuel.

Sinon le nouvel état est accepté malgré tout avec une probabilité $\exp(-\text{diff}/T)$ (règle de Metropolis). Avec :

- *diff*: l'écart entre les deux valeurs de la fonction de coût,
- *T*: une variable qui diminue d'un facteur 0,99 à chaque itération, et dont la valeur de départ est choisie arbitrairement.

Le programme termine après un nombre d'itérations fixé au préalable, ou lorsque la fonction de coût ne change plus de valeur de façon significative.

L'intérêt de cette méthode est qu'elle permet de trouver un minimum global sans rester bloquée dans un minimum local de la fonction de coût.

III. Le support utilisé : Comment modéliser des villes informatiquement ?

Dans une première version de mon TIPE, je modélisais les villes par des graphes dont les points étaient munis de coordonnées, mais il est apparu qu'il fallait utiliser un modèle plus contraignant pour conserver les propriétés d'une ville : en effet la ville doit être connexe, si elle est modélisée par un graphe il doit être planaire, etc.

J'utilise donc un nouveau modèle constitué de cellules, représentant chacune un bloc de bâtiments. Les arrêtes des cellules correspondent aux rues.

- Un **sommet** est un couple d'entiers : ses coordonnées
- Une **cellule** est une liste de sommets correspondant à un cycle. Ses arrêtes sont donc les segments entre deux sommets consécutifs dans la liste. On considère de plus le segment entre le premier élément et le dernier élément de la liste comme une arrête de la cellule, qui vient refermer la boucle.
- Une **ville** est représentée par une liste de cellules.
- Par la suite, le type « ville » comporte 3 sous-éléments : la liste des cellules (**.ce**), la liste des sommets de la ville (**.so**) et la liste des sommets situés au bord (**.bord**).

Quelques règles sont à respecter :

- Toute la surface de la ville est recouverte par les cellules (et il n'y a pas de cellule au-delà).
- Deux cellules ne se coupent pas.
- Une cellule ne contient pas une autre cellule.

Ce dernier modèle est ainsi plus fidèle à la réalité que de simples graphes. Je modélise ainsi seulement la « nouvelle » partie de la ville, laissant un vide au niveau de la ville existant aujourd'hui.

IV. Comment passer d'un état à un état « voisin » ?

Pour passer d'une ville à une ville « voisine », je définis 5 types de modifications élémentaires :

- **Fusionner deux cellules** adjacentes
- **Scinder une cellule** en deux, en reliant deux de ses sommets non adjacents
- **Ajouter un sommet** au centre d'une cellule en la morcelant comme on découperait un gâteau en plusieurs parts.
- **Supprimer un sommet**, en supprimant toutes les arrêtes qui le touchent, c'est-à-dire en fusionnant toutes les cellules qui le touchent
- (Déplacer un sommet, je n'ai pas encore codé cette opération)

Pour obtenir un état voisin de la ville actuelle aléatoire, on tire au hasard une des opérations et on l'applique à un endroit au hasard dans la ville.

La principale difficulté du processus est de s'assurer que ces modifications ne compromettent pas les caractéristiques de la ville : on ne doit pas créer de cellules qui se coupent, déplacer les bords de la surface étudiée, etc.

V. Comment calculer le coût ?

Une fonction convertit la ville en graphe par matrice d'adjacence.

On peut ensuite appliquer l'algorithme de Floyd-Warshall pour obtenir les poids des plus courts chemins entre chaque paire de sommets de la ville. La valeur de la fonction de coût est la somme de tous ces poids.

Par ailleurs on ajoute la longueur totale des routes de la ville dans la fonction de coût, pour s'assurer que la rapidité des trajets ne se fera pas au prix d'un trop grand nombre de routes.

VI. Les points qui posent problème

Beaucoup de facteurs sont arbitraires et devront être ajustés. Notamment :

- Les deux sommes dans la fonction de coût (somme de la longueur des routes et somme des poids des plus courts chemins) devraient être pondérées par des coefficients différents, car

elles n'évoluent pas de la même façon, leur donner le même poids dans la fonction de coût est un choix arbitraire et n'est peut-être pas pertinent.

- La valeur initiale de la température T , le choix de la décroissance selon un facteur 0,99 sont arbitraires.

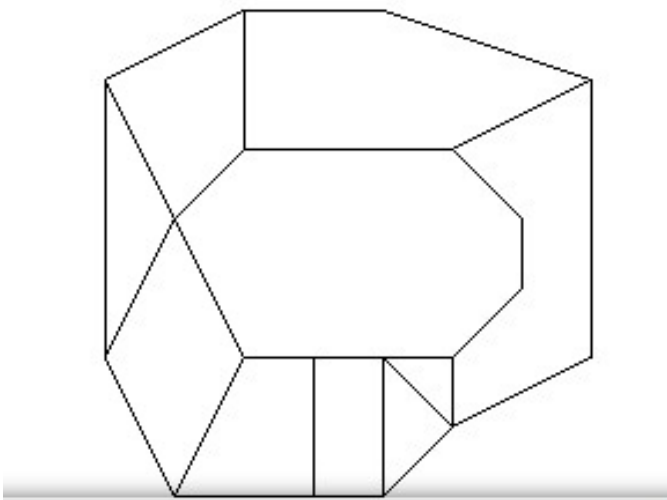
VII. Ce qu'il reste à ajouter

Pour l'instant je ne prends pas en compte les routes existant déjà dans la ville existant aujourd'hui pour calculer les trajets, c'est un élément à ajouter.

La modification « déplacer un sommet » reste à ajouter.

Illustrations :

Exemple de ville codée avec des cellules :



Exemple de ville codée avec un graphe :

