SAE 2.02

Le problème du postier chinois

FILY Ewen
HAZARD Kim — MAIGNAN Anton

Table des matières

Introduction

En théorie des graphes et en algorithmique, le problème du postier chinois, ou problème du postier (en anglais route inspection problème) consiste à trouver un plus court chemin dans un graphe connexe non orienté qui passe au moins une fois par chaque arête et revient à son point de départ. Le nom du problème vient du fait qu'il a été étudié par le mathématicien chinois Meigu Guan en 1962, et qu'il modélise la tournée d'un facteur devant effectuer le plus efficacement possible sa tournée en passant au moins une fois par chaque rue de son secteur. Ce support de table figure un parcours de postier non minimal (6 arêtes dédoublées). Le problème peut être réduit à la recherche d'un couplage maximal de poids minimum, et ainsi être résolu en temps polynomial dans le cas général.

Ce problème permet de répondre à bon nombre de problème d'optimisation concernant le trafic routier, ferroviaire. Permettant de ne pas repasser par des chemins plusieurs fois.

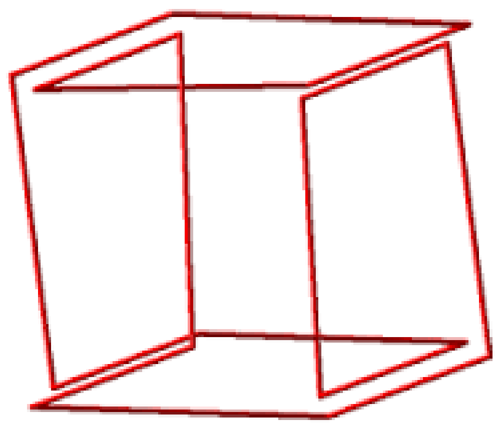


FIGURE 1 – Graphe des arêtes d'un cube (non-eulerien)

Approche

Nous avons commencé par faire nos recherches sur les propriétés d'un graphe eulérien ainsi que la façon de rendre un graphe eulérien. Nous nous sommes concentrés sur la création d'une fonction permettant de rendre le graphe eulérien pour nous permette de poursuivre dans la suite du projet. Les sources utilisées qui ont été utilisées sont Wikipédia et geeksforgeeks.

Demonstration

Euler a démontré les conditions nécessaires en 1735. La démonstration complète ci-dessous ayant été publiée par le mathématicien allemand Carl Hierholzer en 1873. On attribue parfois l'équivalence à Euler, comme dans le livre de théorie des graphes de Diestel.

Supposons qu'il y a un parcours eulérien et empruntons le en supprimant les arêtes utilisées. À chaque passage sur un sommet (sauf au début et à la fin), on supprime l'arête qui arrive sur ce sommet et l'arête qui en part. Ainsi, sauf pour le sommet de départ ou d'arrivée, la parité du degré reste inchangée. À la fin du parcours, toutes les arêtes sont supprimées, ce qui permet de conclure sur la parité des sommets.

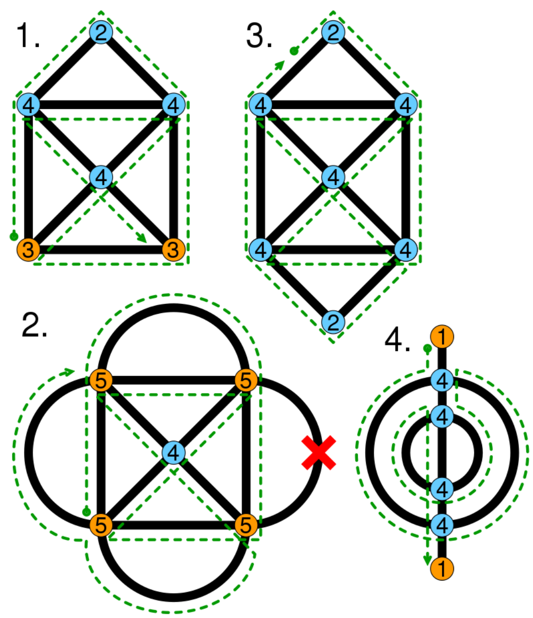


FIGURE 2 – 1. et 4. Graphes ayant un parcours eulérien mais pas de circuit eulérien. 2. Graphe sans solution. 3. Graphe ayant un circuit eulérien.

Algorithme de Fleury

Fonctionnement

Pierre-Henry Fleury a donné un autre algorithme en 1883 mais dont une implémentation sur ordinateur serait moins efficace que l'algorithme de Hierholzer. Son idée est de construire le circuit en empruntant à chaque fois en priorité une arête dont la suppression ne déconnecte pas le graphe.

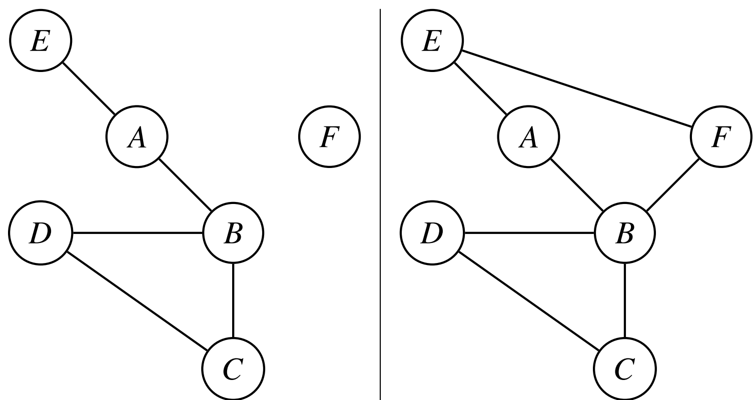
Complexité

La complexité temporelle de l'algorithme de Fleury est peut être écrite comme $O(A^2)$ avec A qui signifie le nombre d'arrêtes dans un graphe non orienté.

La complexité en espace néamoins reste inférieur avec $O(S + A)$ car nous utilisons une liste d'adjacence pour représenter le graphe. La taille de la liste d'adjacence est égale au nombre de sommets plus le nombre d'arêtes dans un graphe non orienté.

Exemple

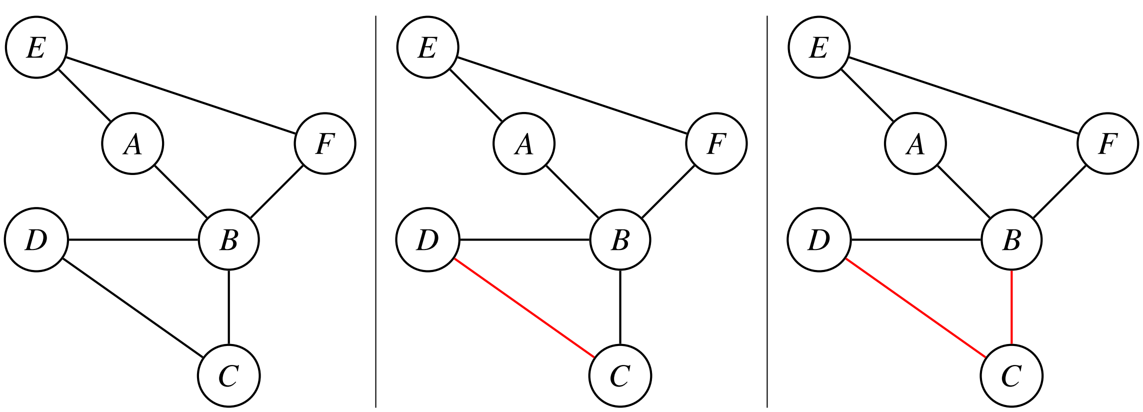
Avant de pouvoir rechercher le chemin eulérien nous devons nous assurer que celui-ci est bien eulérien. Nous utilisons une fonction qui rend eulérien le graphe en ajoutant des arêtes aux sommets isolés (F dans l'exemple ci-dessous) et en ajoutant des arêtes aux sommets ayant des degrés impairs (E et B).

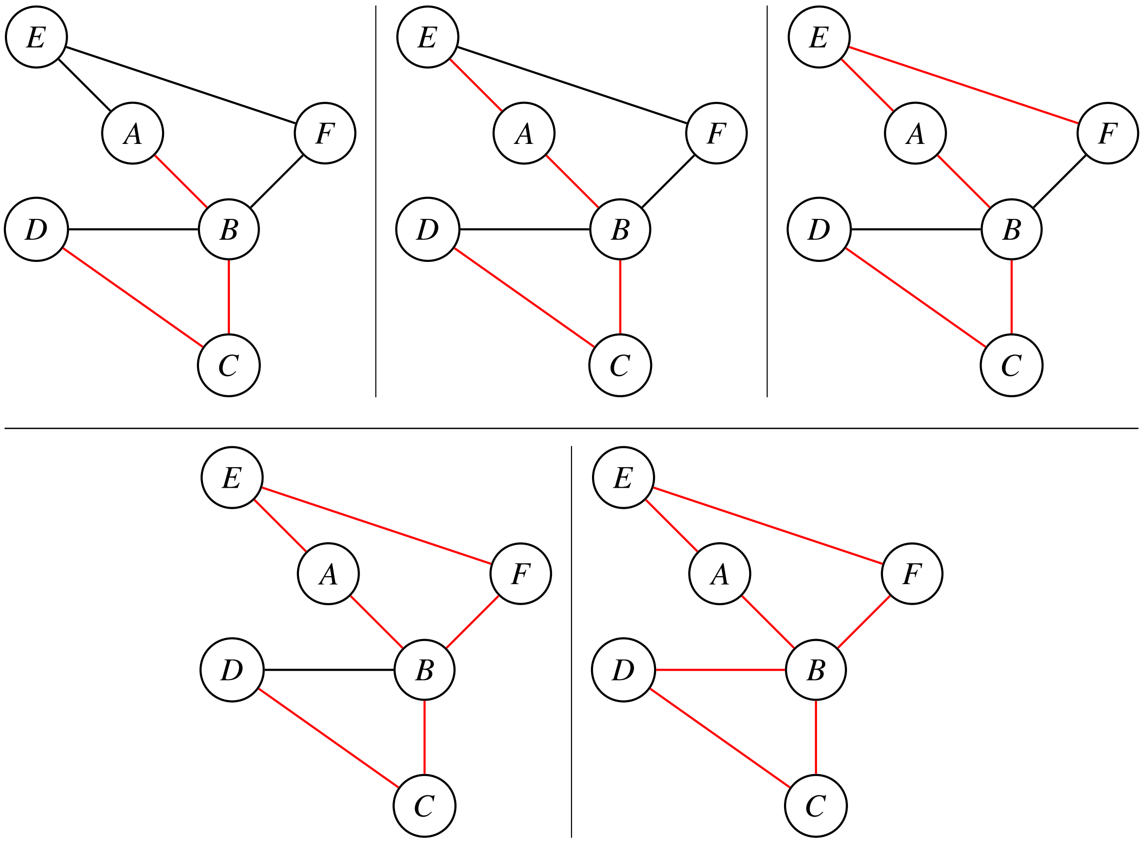


(a) Graphe non eulérien (b) Graphe rendu eulérien

Sachant que le graphe est bien eulérien nous pouvons choisir n'importe quel sommet comme point de départ, prenons D pour l'exemple. Donc le sommet D à deux arêtes. L'algorithme pour choisir l'arête vérifie uniquement si celle-ci n'est pas un **pont** c'est-à-dire une arête dont la suppression déconnecte le graphe. Donc dans ce cas il va choisir la première arête qu'il voit donc $D - C$. Si le sommet possède une seule arête nous empruntons obligatoirement celle-ci.

Une fois l'arête choisit on ajoute dans les sommets traversés D puis nous supprimons du graphe l'arête traversée. On réitère pour C et tous les sommets jusqu'à que le graphe n'ai plus d'arêtes.





Le graphe na plus aucune arête nous pouvons donc afficher le chemin eulérien complet du graphe $D - C - B - A - E - F - B - D$.

Algorithme de Hierholzer

Fonctionnement

On peut effectivement écrire un programme informatique pour calculer un chemin ou un circuit eulérien s'il en existe. Discutons l'algorithme du papier de Hierholzer de 1873, qui suit l'idée de sa démonstration. Il répète l'extraction de circuits que l'on colle pour construire un circuit eulérien. Cet algorithme peut s'implémenter afin d'avoir un algorithme en temps linéaire en le nombre d'arêtes. Pour cela, il suffit que les opérations suivantes s'exécutent en temps constant :

1. trouver une arête non empruntée
2. trouver un nouveau sommet qui admet encore des arêtes incidentes non empruntés
3. coller le circuit dans le parcours en cours de construction

Pour cela, il suffit de maintenir efficacement avec des listes doublement chainées :

1. la liste des sommets du parcours en construction qui ont encore des arêtes inutilisées
2. pour chaque sommet, la liste des arêtes non encore utilisées

3. la liste des arêtes du parcours en construction.

Complexité

La complexité temporelle est $O(S + A)$, où S est le nombre de sommets et A est le nombre d'arêtes du graphe. La raison en est que l'algorithme effectue une recherche en profondeur d'abord (DFS) et visite chaque sommet et chaque arête exactement une fois. Ainsi, pour chaque sommet, il faut $O(1)$ temps pour le visiter et pour chaque arête, il faut $O(1)$ temps pour le parcourir.

La complexité en espace est $O(S + A)$, car l'algorithme utilise une pile pour stocker le chemin actuel et une liste pour stocker le circuit final. La taille maximale de la pile peut être $S + A$ au pire, donc la complexité spatiale est $O(S + A)$.

Exemple

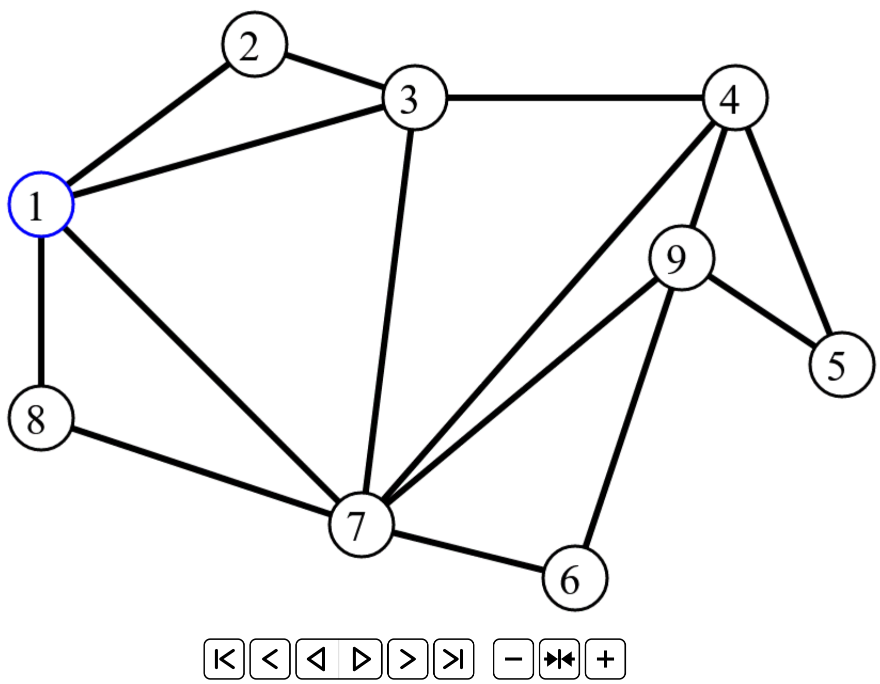


FIGURE 7 – Animation de l'aglrotithme de Hierholzer

Comparaison

L'algorithme de Fleury et l'algorithme de Hierholzer ont des résultats similaires. Nous les avons comparés sur deux graphes composés de 6 sommets et de 7 arêtes et avons enregistré le temps de traitement des deux algorithmes. L'algorithme de Fleury s'en sort en 0.07 secondes et 0.059 secondes tandis que l'algorithme de Hierholzer finit en 0.091 secondes et 0.065 secondes.

Défauts

L'algorithme Hierholzer que nous avons développé n'est pas aussi performant que décris précédemment. Cet algorithme devait être plus rapide en raison de sa complexité plus qui est plus basse que celle de l'algorithme de Fleury.

Améliorations possibles

Nous avons deux grosses améliorations possible pour répondre au mieux au problème du postier chinois.

1. Améliorer la fonction qui rend Eulérien le graphe. Notre fonction rendant eulérien un graphe possèdent des limites comme le fait qu'elle ne peut qu'ajouter des arêtes et n'en supprimer pas . De plus un graphe ne peut être eulérien si les sommets sont impairs. Nous pourrions ajouter des sommets afin d'avoir un circuit eulérien complet.
2. Optimiser l'algorithme de Hierholzer. Malgré nos efforts nous avons pas pu atteindre la complexité visée de $O(S + A)$.

Conclusion

Le problème du postier chinois est un parfait exemple de problème qui a su rester d'actualité, utilisé dans la vérification des algorithmes de réseau pour le rabotage des chasse-neige ou l'entretien des rues. D'autre algorithme sont aujourd'hui utilisés comme CPT (Compact Prediction Tree) pour analyser le réseau neuronal.