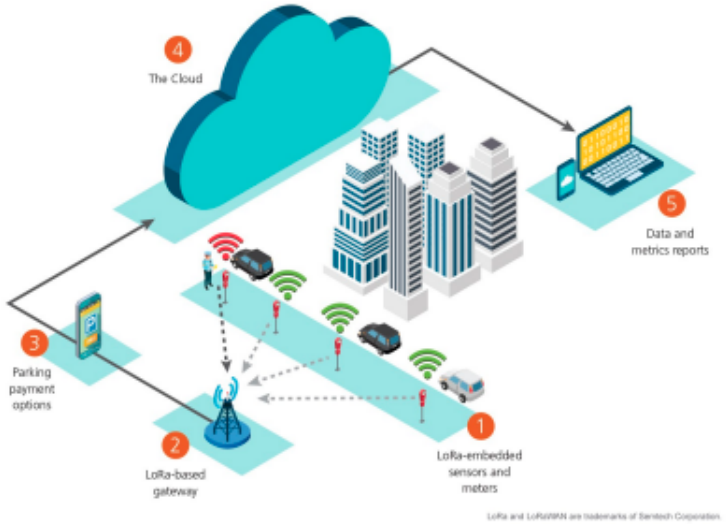
INPT
المعهد الوطني للبريد والمواصلات
Institut National des Postes et Télécommunications

Rapport de projet

COLORATION DES GRAPHES



Réalisé par :
AKEL Rihab
SEDATI Ilham
ELKORRI Hanaa

Encadré par Mr BENSAID
Hichame

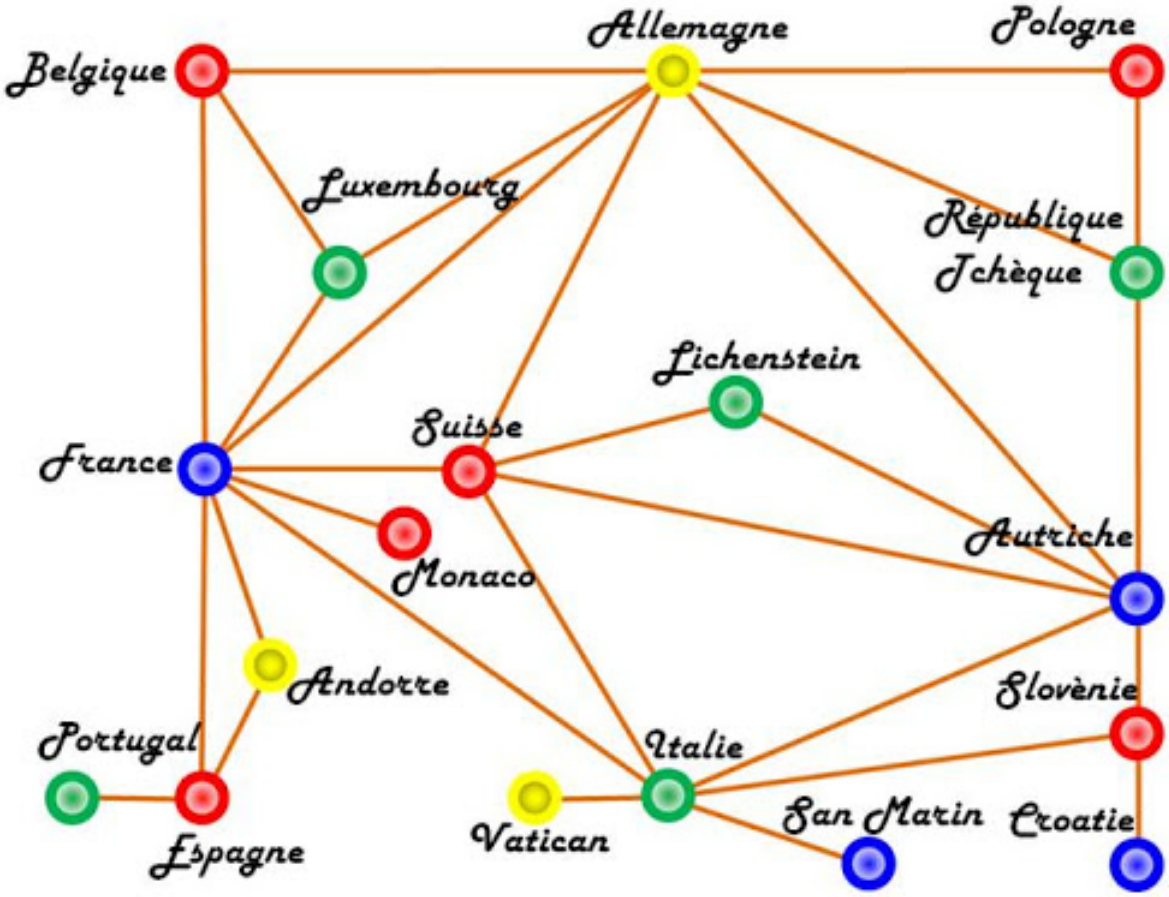
Filière SUD - CLOUD & IOT - INE1

Table des matières

0.1 Introduction

En théorie des graphes, la coloration de graphe consiste à attribuer une couleur à chacun de ses sommets de manière que deux sommets reliés par une arête soient de couleur différente. On cherche souvent à utiliser le nombre minimal de couleurs, appelé nombre chromatique qui est le minimum de couleurs différentes nécessaires pour colorier tous les sommets d'un graphe de façon à ce que deux sommets adjacents aient des couleurs différentes. Le champ d'applications de la coloration de graphe couvre notamment le problème de l'attribution de fréquences dans les télécommunications, la conception de puces électroniques ou l'allocation de registres en compilation…

Graphe coloré pour l'Europe occidentale :

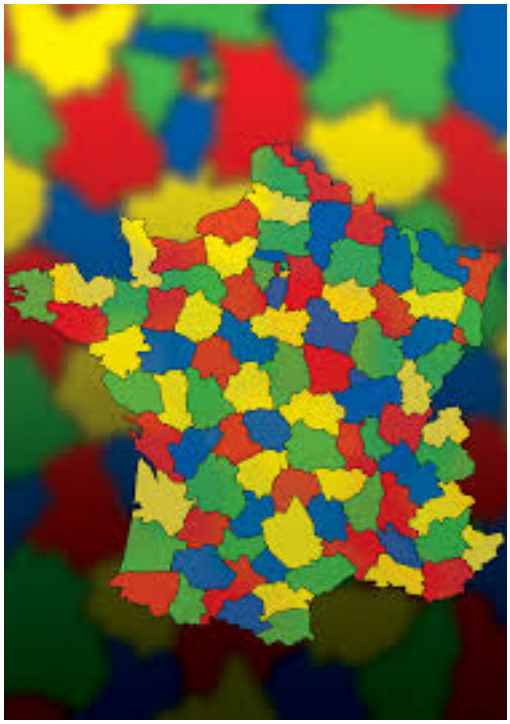


0.2 Historique

Les premiers résultats de coloration de graphe concernent presque exclusivement les graphes planaires (un graphe planaire est un graphe qui a la particularité de pouvoir se représenter sur un plan sans qu'aucune arête (ou arc pour un graphe orienté) n'en croise une autre. Autrement dit, ces graphes sont précisément ceux que l'on peut plonger dans le plan. : il s'agissait alors de colorier des cartes.

En cherchant à mettre en couleurs une carte des comtés d'Angleterre, Francis Guthrie postula la conjecture des quatre couleurs. Il remarqua en effet qu'il n'y avait besoin que de quatre couleurs pour que deux comtés ayant une frontière commune soient de couleurs différentes. Le frère de Guthrie transmit la question à son professeur de mathématiques, Auguste de Morgan de l'University College de Londres. De Morgan mentionna ce problème dans une lettre adressée à William Rowan Hamilton en 1852. Arthur Cayley évoqua cette question lors d'un colloque de la London Mathematical Society en 1879. La même année, Alfred Kempe publia ce qu'il prétendit en être une démonstration et pendant une décennie, on crut que le problème des quatre couleurs était résolu. Kempe fut élu membre de la Royal Society et devint ensuite président de la London Mathematical Society.

En 1890, Percy John Heawood fit remarquer que la démonstration de Kempe était fausse. Il montra quant à lui le théorème des cinq couleurs en reprenant des idées de Kempe. De nombreux travaux ont été publiés lors du siècle suivant pour réduire le nombre de couleurs à quatre, jusqu'à la démonstration finale de Kenneth Appel et Wolfgang Haken. La preuve réutilisait les idées d'Heawood et Kempe et pratiquement pas les développements ultérieurs. Il s'agit aussi de la première preuve majeure utilisant massivement l'ordinateur.



Les premiers résultats de coloration de graphe concernent presque exclusivement les graphes planaires : il s'agit alors de colorier des cartes. Des mathématiciens ont démontré, dès la fin du 19e siècle, qu'il suffisait de 4 couleurs pour colorier une carte géographique

0.3 Problèmes algorithmiques

• Problème de décision : étant donné G, un graphe, et k un entier, existe-t-il une coloration valide de G utilisant k couleurs ?

• Problème d'optimisation : étant donné G, quel est son nombre chromatique ?

0.4 Algorithmes

0.4.1 Heuristiques

L'importance du problème a donné lieu à l'élaboration de nombreuses heuristiques spécifiques au problème, spécialement des algorithmes séquentiels de coloration sommet par sommet. Elle a aussi suscité de nombreuses expérimentations des méthodes approchées générales : méta-heuristique (recuit simulé, recherche tabou), ou encore algorithme génétique.

0.4.2 Algorithme de Welsh et Powell

1. Repérer le degré de chaque sommet. 2. Ranger les sommets par ordre de degrés décroissants (dans certains cas plusieurs possibilités).

3. Attribuer au premier sommet (A) de la liste une couleur.

4. Suivre la liste en attribuant la même couleur au premier sommet (B) qui ne soit pas adjacent à (A).

5. Suivre (si possible) la liste jusqu'au prochain sommet (C) qui ne soit adjacent ni à A ni à B.

6. Continuer jusqu'à ce que la liste soit finie.

7. Prendre une deuxième couleur pour le premier sommet (D) non encore coloré de la liste.

8. Répéter les opérations 3 à 7.

9. Continuer jusqu'à avoir coloré tous les sommets.

Remarquons que cette méthode peut aboutir à la pire des colorations possibles, par-exemple si le graphe G a la structure de couronne à n sommets, son nombre chromatique est 2 (si n est pair) tandis que Welsh-Powell donne dans certains cas (selon l'ordre dans lequel sont rangés les sommets) une coloration utilisant n/2 couleurs ! L'heuristique DSATUR permet d'éviter ce problème et donne une coloration moins mauvaise dans le pire cas.

0.4.3 DSATUR

DSATUR est une heuristique qui consiste à colorer les sommets les uns après les autres, en s'appuyant sur un tri préalable des sommets. Une priorité est donnée aux sommets de grand degré, ainsi que les sommets dont les voisins ont déjà obtenu le plus couleurs différentes.

0.4.4 Algorithme de 2-coloriage

On considère un graphe G=(S,A) 2-coloriable.

Pour chaque composante connexe de G on se donne un sommet s appartenant à S on lui attribue la couleur 0 et on attribue la couleur 1 à ses voisins. Pour chacun de ses voisins, on attribue la couleur 0 à ses voisins non coloriés, et ainsi de suite : on réalise ainsi un parcours en largeur du graphe, en coloriant les sommets rencontrés de la façon énoncée précédemment.

Correction de l'algorithme

L'algorithme de 2-coloriage renvoie bien un coloriage si le graphe en entrée est 2-coloriable. En effet, si on prend 2 sommets voisins, l'un des sommets a été parcouru le premier. Le deuxième sommet est donc colorié de l'autre couleur par l'algorithme, et sa couleur n'est pas modifiée par la suite. De plus, l'algorithme effectue un parcours en largeur, donc tous les sommets sont coloriés.

0.4.5 Algorithme de Wigderson

L'algorithme de Wigderson est un algorithme de complexité en temps polynomiale, qui colore avec $O(\sqrt{n})$ couleurs les graphes 3-coloriables.

Cet algorithme s'effectue sur des graphes qu'on sait 3-coloriables. Soit G=(S,A) un tel graphe. On note n le nombre de sommets de G.

1. On prend comme couleur initiale c=0 ;

2. Pour tout s appartenant à S non déjà colorié et avec au moins $\sqrt{n}$ voisins non coloriés : on considère le sous graphe induit par l'ensemble des voisins de s pas encore coloriés, et on le 2-colorie avec les couleurs c et c+1 On ajoute à c le nombre de couleurs utilisées.

3. Avec l'algorithme glouton, on colorie le reste des sommets avec des couleurs plus grandes que c .

La complexité de l'algorithme de Wigderson est polynomiale en n et détermine un coloriage de G qui utilise $O(\sqrt{n})$ couleurs.

0.4.6 Algorithme distribué

La coloration est un problème classique de calcul distribué synchrone. Un résultat notable13 est la borne inférieure de Nati Linial : il faut $\Omega(\log^* n)$ étapes pour faire une 3 coloration d'un cycle de longueur (même pour un algorithme utilisant de l'aléa), où log* est le logarithme itéré.

0.5 Algorithme Naïf

Approche

Cet algorithme continue de colorer tous les sommets du graphe jusqu'à ce qu'il atteigne la taille du graphe. donc quand la taille est atteinte il appelle la "fonction issafe" pour vérifier, au cas où ce n'est pas juste l'algorithme ajoute une autre couleur à la liste des couleurs et recommence à zéro.

Algorithme

Explication de l'algorithme

> Dans la fonction "main" nous demandons à l'utilisateur d'entrer la taille du graphe, et tant que la taille n'est pas nulle nous appelons la fonction "graphcoloring".

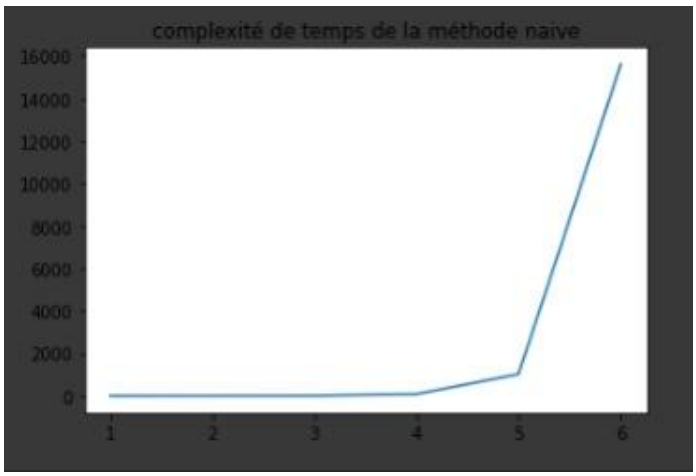
> "graphcoloring" est une fonction récursive qui retourne faux si la coloration n'est pas possible. sinon elle retourne vrai et assigne des couleurs à tous les sommets

> "issafe" est une fonction qui vérifie si le graphe coloré est sûr ou non, en vérifiant si les sommets adjacents ont la même couleur ou non.

> à la fin la fonction "printsolution" qui retourne une liste de couleurs pour chaque sommet.

Etude de la complexité

Il existe une combinaison totale de couleurs O (m^V). La complexité temporelle est donc O (m^V) .



0.6 Algorithme Glouton

Solution

car il n'y a pas d'algorithme efficace disponible pour colorer un graphe avec un nombre minimum de couleurs car le problème est un problème NP-Complete connu, nous prenons donc les algorithmes gloutons comme une solution optimales qui est utilisés pour assigner des couleurs aux vétices mais ne garantissent pas d'utiliser des couleurs minimum, mais il garantit une limite supérieure sur le nombre de couleurs.

Approche

Dans cette algorithme nous colorions le premier sommet avec la première couleur. et pour le reste, nous choisissons un sommet et le colorions avec le couleur numérotée la plus basse qui n'a été utilisée sur aucune autre sommets colorés adjacents. Si toutes les couleurs utilisées précédemment apparaissent sur les sommets adjacents au sommet actuel, nous lui attribuons une nouvelle couleur.

Algorithme

> On initialise tous les sommets sur l'état incolore.

> Puis on attribue la première couleur au premier sommet.

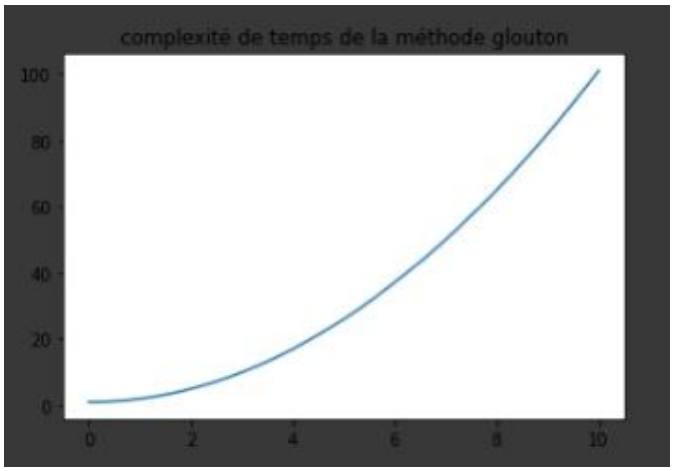
> Puis tester si le sommet adjacent au sommet actuel s'il est coloré, si oui, nous renvoyons sa valeur comme invalide.

> Après nous cherchons une autre couleur et nous l'affectons au sommet actuel.

> Et à la fin, nous retournons un graphique en couleur.

Etude de la complexité

Complexité temporelle : O (V^2) dans le pire des cas.



0.7 Conclusion

En se concentrant sur la complexité de l'algorithme gourmand, on remarque que sa complexité est optimale que celle du naïf. d'où l'on peut conclure que même si les algorithmes gourmands ne nous garantissent pas un nombre minimum de couleurs pour colorer le graphe, car le nombre de couleurs utilisées dépend parfois de l'ordre des sommets et de la manière dont ils sont traités, mais il garantit une limite supérieure du nombre de couleurs.