Cours de modélisation mathématique

Guilherme Dias da Fonseca

TP 5 – Coloration

1 Présentation abstraite

Entrée : Un graphe non-dirigé $G = (V, E)$, où V est l'ensemble de sommets et E est l'ensemble d'arêtes.

Sortie : Une fonction $f : V \to \{0, \ldots, k-1\}$ telle que si $uv \in E$, alors $f(u) \neq f(v)$.

Objectif : Minimiser le *nombre de couleurs* k.

En d'autres termes, on veut trouver une partition de V parmi k ensembles indépendants et minimiser k.

2 Présentation appliquée (source : wikipedia)

Certains réseaux de télécommunication sont composés d'émetteurs émettant chacun sur une fréquence particulière. Lorsque deux émetteurs sont trop proches on ne peut leur allouer la même fréquence à cause des interférences (sauf si éventuellement une montagne les sépare).

On associe un graphe au réseau—chaque sommet est un émetteur et chaque arête spécifie que l'on ne veut pas allouer la même fréquence aux deux émetteurs correspondant à ses deux extrémités—et ainsi déterminer une allocation réalisable avec un minimum de fréquences (dont la licence d'exploitation peut entrainer un coût important). Ce problème est un cas particulier du problème de la coloration de graphe.

Précisons que les contraintes technologiques dépendant du relief géographique, on ne peut pas faire l'hypothèse qu'un graphe obtenu à partir des réseaux de télécommunications soit un graphe de disques.

3 Exemple

Disons que l'entrée est le graphe avec l'ensemble de sommets

$$V = \{0, 1, 2, 3, 4, 5, 6, 7\}$$

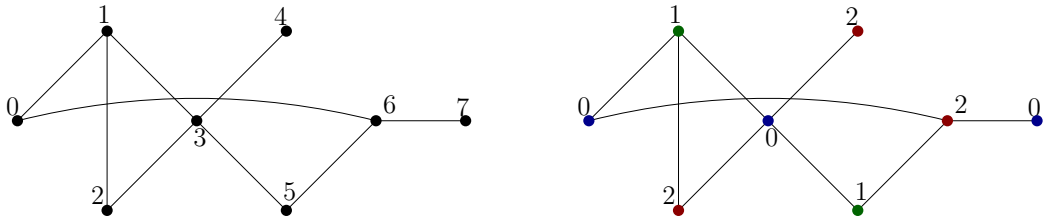

FIGURE 1 – Exemple de graphe et sa coloration avec 3 couleurs.

et l'ensemble d'arêtes

$$V = \{\{0,1\},\{0,6\},\{1,2\},\{1,3\},\{2,3\},\{3,4\},\{3,5\},\{5,6\},\{6,7\}\}.$$

Une coloration en utilisant 3 couleurs est montrée sur la figure 1.

4 Fichiers d'entrée

Le fichier d'entrée est textuel et la plupart des lignes contient deux entiers séparés par un espace, qui indiquent qu'il y a une arête entre les deux sommets. La première ligne contient le nombre de sommets n (qui sont numérotés de 0 à $n-1$) et la deuxième ligne contient le nombre d'arêtes m. Chaque arête apparait seulement une fois dans le fichier, soit dans une direction, soit dans l'autre.

Pour l'exemple on a le fichier :

```
8
9
0 1
0 6
1 2
1 3
2 3
3 4
3 5
5 6
6 7
```

Il y a 4 fichiers d'entrée sur l'ENT :

1. `10.in5` : $n = 10, m = 15$
   Graphe de Petersen (figure 2).
2. `100.in5` : $n = 100, m = 974$
   Graphe aléatoire où la probabilité de choisir chaque arête est égale à 20%.
3. `1000.in5` : $n = 1000, m = 2980$
   Triangulation de Delaunay de 1000 villes grecques (figure 3).
   Par le Théorème des quatre couleurs, on sait qu'on peut le colorer avec 4 couleurs.
4. `4039.in5` : $n = 4039, m = 87889$
   Relations d'amitié sur facebook.
   Source : `http://snap.stanford.edu/data/egonets-Facebook.html`

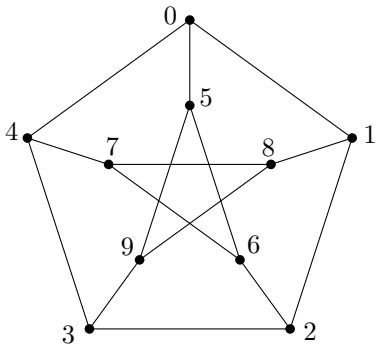

FIGURE 2 – Graphe de Petersen `petersen-10.edg`.

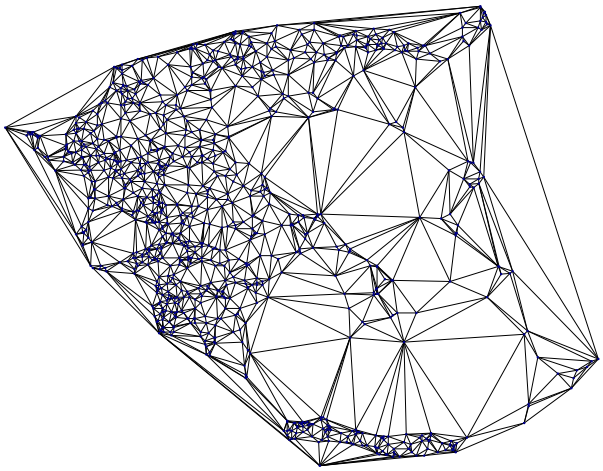

FIGURE 3 – Graphe du fichier `1000.in4`, correspondant à la triangulation de Delaunay de 1000 villes grecques.

5 Fichier de sortie

La première ligne du fichier de sortie est l'entier n (nombre de sommets) et la deuxième est l'entier k (nombre de couleurs). Ensuite on liste la couleur des n sommets en ordre, une couleur par ligne. Pour l'exemple on a le fichier :

```
8
3
0
1
2
0
2
1
2
0
```

6 Stratégies

Récursive On choisi un sommet v (de préférence avec peu de voisins). Ensuite, on enlève v du graphe et on fait un appel récursif pour colorer le reste du graphe. Une fois qu'on obtient le résultat, on remet v dans le graphe et on colore v avec le plus petit entier que n'est pas utilisé par ses voisins. La base de la récursion est colorer un graphe d'un seul sommet. Il faut augmenter la limite d'appels récursifs en python avec `sys.setrecursionlimit(nombre)`.

Itérative On choisi une ordre pour les sommets (plus de voisins d'abord ?). Pour chaque sommet v en ordre, on colore v avec le plus petit entier que n'est pas encore utilisé par ses voisins. On répète jusqu'à avoir coloré tous les sommets.

Itérative répétée On exécute la stratégie itérative et on trouve les sommets problématiques. On dit qu'on sommet est problématique si il utilise une couleur pas encore utilisée. On place les sommets problématiques un peu plus tôt dans la liste et on répète la stratégie itérative.

DSATUR On peut utiliser la heuristique DSATUR `http://fr.wikipedia.org/wiki/DSATUR`.

Local search On peut changer les couleurs des sommets pour essayer d'éliminer une couleur.

7 Notation

La notation sera basée sur le tableau ci-dessous, qui indique le nombre de couleurs nécessaire pour avoir la note à gauche pour chaque fichier. La note du projet est la moyenne des notes des 4 fichiers.

Note	`10.in5`	`100.in5`	`1000.in5`	`4039.in5`
10	5	11	7	76
12				75
14	4	10	6	73
16		9		72
18		8	5	71
20	3	7	4	70