Coloration de graphes

Paul Melotti

juin 2014

Un *graphe* G est un couple (V, E) où V est un ensemble fini d'éléments appelés *sommets*, et E un ensemble de paires non ordonnées (v, w) où $v, w \in V$ et $v \neq w$, appelées arêtes. Deux sommets sont dits *adjacents* s'il existe une arête les reliant.

Le problème du coloriage de graphe consiste à assigner à chaque sommet une couleur, de façon à ce que deux sommets adjacents soient de couleurs différentes. Si c'est possible avec k couleurs, on dit que le graphe est *k-coloriable*.

1. Quelle structure informatique utiliser pour représenter un graphe ?

1 2-coloriages

Le graphe G est dit *biparti* si V se partitionne en deux ensembles B et W, tels que toute arête de E relie un sommet de B à un sommet de W.

2. Montrer qu'un graphe est 2-coloriable si et seulement si il est biparti.
3. Donner un algorithme qui, lancé sur un graphe biparti, détermine une 2-coloration de ce graphe. Quelle est sa complexité ? La réponse peut dépendre de la structure de donnée considérée.
4. Montrer qu'un graphe est 2-coloriable si et seulement si il ne possède pas de cycle de longueur impaire.

2 Algorithme glouton

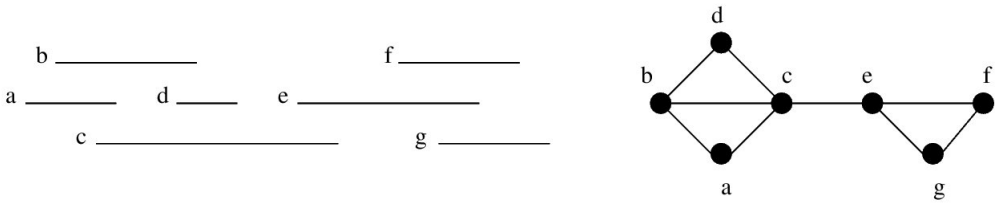
Le problème du 2-coloriage est donc "facile". On va s'intéresser maintenant aux n-coloriages pour $n \geq 3$, ce qui est beaucoup plus difficile. Le problème consiste à colorier le graphe avec un nombre minimal de couleurs.

On représente les couleurs par les entiers $1, 2, \ldots, K, \ldots$. On appelle *degré* d'un sommet v le nombre de ses voisins, et on le note $\delta(v)$. On appelle degré du graphe G le maximum des $\delta(v)$ pour $v \in V$, et on le note $\Delta(G)$.

L'algorithme glouton consiste à trouver un coloriage du graphe de la façon suivante : on numérote les sommets $v_1, \ldots, v_n$ de G, puis on les colorie dans cet ordre en attribuant à chaque sommet la plus petite couleur disponible (*i.e.* non déjà utilisée par ses voisins). On appelle $gl(G)$ le nombre total de couleurs utilisées par cet algorithme.

5. Montrer que $gl(G) \leq \Delta(G) + 1$.
6. Montrer qu'on a en fait $gl(G) \leq \max_{i=1}^{n} \min(\delta(v_i) + 1, i)$.
7. Quelle numérotation des sommets cette inégalité suggère-t-elle d'utiliser ?

L'algorithme glouton est optimal pour une famille de graphes, les *graphes d'intervalles*. Étant donné une famille d'intervalles sur la droite réelle, on définit un graphe dont les sommets sont les intervalles et dont les arêtes relient les sommets représentant des intervalles qui s'intersectent. Exemple :



On numérote les sommets de ce graphe selon l'ordre donné par les extrémités gauches des intervalles. Dans notre exemple, cela donne a, b, c, d, e, f, g.

8. Faire tourner l'algorithme à la main sur cet exemple
9. Montrer que cet algorithme est optimal (*i.e.* utilise le nombre minimal de couleurs) pour tout graphe d'intervalles.

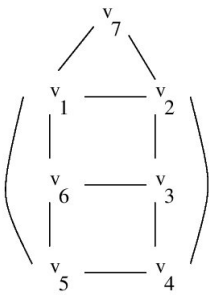
3 Algorithme de Brelaz

On propose ici une variante de l'algorithme précédent. À un moment quelconque de l'exécution de l'algorithme, on appelle *degré-couleur* d'un sommet le nombre de couleurs déjà utilisées pour colorier ses voisins (attention, ce n'est pas le nombre de voisins déjà coloriés : certains peuvent avoir la même couleur). Initialement, le degré-couleur de chaque sommet est nul puis il évolue jusqu'à ce que le graphe soit colorié. Plus précisément, l'algorithme de Brelaz fonctionne de la façon suivante :

Initialisation : trier les sommets par degré décroissant. Choisir un sommet de degré maximal et lui assigner la couleur 1.

Régime permanent : tant qu'il reste des sommets non coloriés, choisir un sommet non colorié de degré-couleur maximal, et en cas d'égalité prendre parmi ceux-ci un sommet de degré maximal. Le colorier avec la plus petite couleur admissible.

10. Faire tourner l'algorithme à la main sur l'exemple suivant :



11. Montrer que sur un graphe biparti, l'algorithme de Brelaz n'utilise que deux couleurs. Est-ce le cas de l'algorithme glouton vu précédemment ?

Les algorithmes que nous avons vus sont polynomiaux, mais ils peuvent utiliser un nombre non-optimal de couleurs. On ne connaît pas d'algorithme polynomial qui trouve un coloriage optimal dans le cas général, c'est un problème NP-complet.