Master2 Informatique
UE: Algorithmique Distribuées – 4TIN913U
Responsable : M. Gavoille
Date : 13 janvier 2021
Durée : 1h30
Cours et notes de cours papier autorisés

---

# 1 Aller (enfin) au cinéma

Un groupe d'étudiants souhaitent aller voir un film tous ensemble. Chacun a une idée du film qui veut voir. L'objectif est d'arriver à se mettre d'accord sur le titre du film permettant de choisir une séance. Bien sûr personne ne veut aller voir un film que personne ne voulait aller voir. Ici « choisir » ou « se mettre d'accord » signifie qu'au bout d'un certain temps, chaque étudiant a réussit à choisir définitivement la séance commune. Les communications ont lieu exclusivement via une messagerie instantanée, un réseau social type discord, où chacun peut communiquer avec tous les autres sans qu'il y perte de message. Si écrire ou lire sur ce réseau peut être considéré comme instantané, on ne peut pas faire d'hypothèse particulière sur la vitesse de réaction de chacun ce qui confère au réseau un caractère fondamentalement asynchrone. De plus, chaque étudiant peut se désister de manière passive, c'est-à-dire ne pas répondre (ne serait-ce que pour des raisons techniques de connexions) sans le revendiquer (il ne signale pas forcément qu'il ne participe plus).

**Question 1.** *À quel problème vu au début du cours se réduit ce problème si le choix est entre les films « Imitation Game » ou « Interstellar » ?*

**Question 2.** *Quelle solution pourrait-on envisager ? Justifiez.*

# 2 Coloration par niveau

Dans cette partie, on considèrera exclusivement des algorithmes distribués déterministes de coloration s'exécutant dans le modèle LOCAL. Comme dans le cours, pour un graphe $G$ on note $\Delta = \max\{\deg(u) : u \in V(G)\}$ le degré maximum du graphe et $n$ son nombre de sommets.

**Question 3.** *Dans le problème de la $(\Delta + 1)$-coloration, quelle est la motivation pour le choix du nombre de couleurs ? C'est-à-dire pourquoi avoir choisi $\Delta + 1$ couleurs plutôt que $\lceil \Delta/2 \rceil$ couleurs par exemple ?*

Soit $f(\Delta, n)$ la complexité en temps (le nombre de rondes) d'un algorithme distribué rapide permettant de produire pour tout graphe $G$ à $n$ sommets une $(\Delta + 1)$-coloration.

**Question 4.** *Sans le justifiez, donnez la borne supérieure vue en cours sur $f(\Delta, n)$.*

On va introduire maintenant une nouvelle méthode de coloration s'appliquant aux graphes ayant une certaine décomposition. Plus précisément, une $(k, t)$-*décomposition* d'un graphe $G$ est une suite $(G_1, G_2, \ldots, G_t)$ de $t$ sous-graphes induits de $G$ telle que :

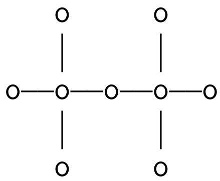
chaque sommet $u$ de $G$

- apparaît dans un unique sous-graphe $G_i$ de la décomposition, et on dira que $u$ est de *niveau* $i$.
- possède au plus $k$ voisins de niveau supérieur ou égal à son propre niveau.

Pour simplifier on notera $G_i^+$ le sous-graphe de $G$ induit par tous les sommets de niveau $\geqslant i$. Bien évidemment $G_1^+ = G$.

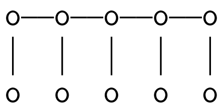
Une façon (séquentielle) d'obtenir une $(k,t)$-décomposition de $G$, si elle existe, est de placer dans $G_1$ tous les sommets de $G_1^+$ $(= G)$ qui ont un degré $\leqslant k$, puis de les supprimer pour obtenir le sous-graphe $G_2^+$. On recommence ensuite en supprimant tous les sommets de degré $\leqslant k$ de $G_2^+$, pour les placer dans $G_2$, et obtenir $G_3^+$. Et ainsi de suite, $t$ fois, jusqu'à obtenir la séquence complète $(G_1, \ldots, G_t)$. En pratique on pourra s'arrêter lorsqu'on tombe sur un sous-graphe $G_i^+$ vide car si $G$ possède une $(k,t)$-décomposition[1] $(G_1, \ldots, G_{t-1}, \varnothing)$, alors $(G_1, \ldots, G_{t-1})$ sera une $(k, t-1)$-décomposition de $G$.

**Question 5.** *Calculer une $(1,3)$-décomposition de l'arbre ci-dessous, en reportant simplement le niveau de chacun des sommets.*



**Question 6.** *Quelle condition doit satisfaire un graphe $G$ (à $n$ sommets) pour qu'il possède une $(k,n)$-décomposition ?*

On considère le graphe *peigne* à $n$ sommets, en supposant $n$ pair, le graphe formé d'un chemin de $n/2$ sommets chacun ayant un sommet pendant (de degré 1). Ci-dessous est dessiné un peigne à 10 sommets.



**Question 7.** *Que vaut le nombre minimum de niveaux $t$ dans une $(1,t)$-décomposition du peigne à 10 sommets ?*

**Question 8.** *De manière générale, que vaut le nombre minimum de niveaux $t$ dans une $(1,t)$-décomposition d'un peigne à $n$ sommets ?*

**Question 9.** *Que vaut le nombre de niveaux minimum $t$ dans une $(2,t)$-décomposition d'un peigne à $n$ sommets ?*

**Question 10.** *Supposons que $G$ est un graphe à $n$ sommets et $m$ arêtes, et qu'il existe deux entiers $k, b \geqslant 1$ tels que $m \leqslant (k+1) \cdot (1 - 1/b) \cdot n/2$. Montrez que $G$ possède au moins $n/b$ sommets de degré $\leqslant k$.*

Dans toute la suite, vous pourrez considérer que la propriété de la question 10 est vraie.

**Question 11.** *En déduire que tout arbre à $n$ sommets possède une $(3, \lceil \log_2 n \rceil - 1)$-décomposition et une $(2, \lceil \log_{3/2} n \rceil - 1)$-décomposition. [Aide : Pensez que la décomposition peut s'arrêter lorsqu'il ne reste qu'au plus $k+1$ sommets. Pourquoi ?]*

On va maintenant présenter une méthode pour obtenir une $(k+1)$-coloration d'un graphe $G$ possédant une $(k,t)$-décomposition $(G_1, \ldots, G_t)$. On supposera ici que $k$ et $t$ sont des paramètres connus, et que dans une $(k+1)$-coloration, les couleurs sont des entiers de $[0,k]$. On utilisera la fonction FirstFree vue en cours définie par $\text{FirstFree}(X) = \min(\mathbb{N} \setminus X)$, pour tout $X \subseteq \mathbb{N}$.

Les grandes étapes de la méthode sont :

1. Calculer le niveau de chacun des sommets de $G$.
2. Calculer en parallèle une $(k+1)$-coloration de chaque $G_i$.

1. On note $\varnothing$ le graphe vide, sans sommet donc.

3. En partant de $i = t-1, \ldots, 2, 1$, re-colorier chaque sommet de $G_i$, grâce à FirstFree appliquée par couleur et sur le graphe $G_i^+$.

Cette méthode donne bien une $(k+1)$-coloration en remarquant que chaque sommet $u$ de niveau $i$ qui se re-colorie est de degré $\leqslant k$ dans $G_i^+$ laissant une couleur libre pour $u$. De plus, on peut vérifier qu'il n'y a jamais deux voisins de $G$ qui se re-colorient à la même ronde.

L'objectif est maintenant de donner une implémentation distribuée de cette méthode. On supposera qu'un sommet ne connaît pas directement son degré dans le graphe $G$, mais qu'il peut l'apprendre en communiquant avec ses voisins, par exemple en comptant le nombre de messages reçus après une ronde.

**Question 12.** *Donnez un algorithme distribué, au moins dans ses principes, permettant d'implémenter l'étape 1 de la méthode, c'est-à-dire de calculer pour chaque sommet $u$ de $G$ son niveau qu'on notera $\ell(u)$. Précisez le nombre de rondes.*

**Question 13.** *Donnez un algorithme distribué, au moins dans ses principes, permettant d'implémenter l'étape 2 de la méthode. En utilisant la fonction $f$, précisez le nombre de rondes.*

**Question 14.** *Détaillez l'implémentation distribuée pour l'étape 3 en précisant le nombre de rondes obtenues.*

**Question 15.** *En déduire le nombre total de rondes pour implémenter la méthode de $(k+1)$-coloration.*

**Question 16.** *Déduire des questions précédentes qu'on peut calculer en $O(\log n)$ rondes une 3-coloration pour tout arbre à $n$ sommets. Justifiez.*

**Question 17.** *Cette dernière coloration, est-elle un progrès d'après l'algorithme distribué de 3-coloration en $O(\log^* n)$ rondes vu en cours ? Justifiez.*

**Question 18.** *Pour les grilles carrées à $n$ sommets, donnez une $(3,t)$-décomposition avec le $t$ minimum. Quel serait le $t$ minimum pour une $(2,t)$-décomposition ?*

**Question 19.** *Montrez que tout graphe planaire à $n$ sommets possède un algorithme distribué de 7-coloration en $O(\log n)$ rondes.*

**FIN.**