Notes de cours – CM7

1 Algorithmes efficaces pour des classes structurées

La plupart des problèmes abordés dans ce cours sont NP-complets, c'est-à-dire difficiles à résoudre de manière exacte en un temps raisonnable. Cependant, leur difficulté provient parfois uniquement de quelques instances ayant une structure compliquée, qui ne se retrouve pas forcément dans des instances "naturelles" de la vraie vie. L'objectif de cette partie est de montrer comment obtenir des algorithmes efficaces lorsqu'on se restreint à des classes de graphes mieux structurées, et dont la structure découle naturellement de la modélisation des problèmes considérés.

1.1 Graphes bipartis

On rappelle qu'un graphe biparti est un graphe dont on peut partitionner les sommets en deux ensemble indépendants (il ne peut y avoir des arêtes qu'entre les deux parties). De manière alternative, il s'agit des graphes ne contenant pas de cycles impairs.

Par définition, colorer un graphe biparti est facile : ils sont tous 2-colorables. De manière similaire, mais un peu plus élaborée, on peut résoudre facilement la plupart des problèmes rencontrés jusqu'à maintenant.

Théorème 1. *Dans les graphes bipartis, on peut résoudre en temps polynomial les problèmes de couplage, couverture de sommets et ensemble indépendant.*

Pourquoi vouloir trouver un couplage dans un graphe biparti ? → problème de mariages.

Pour montrer ce théorème, on utilise un résultat classique de König :

Théorème 2. *Dans un graphe biparti, la taille μ du couplage maximum est égale à la taille τ de la couverture de sommets minimum.*

Démonstration. On a clairement $\mu \leqslant \tau$ car chaque arête d'un couplage doit contenir au moins un sommet de la couverture (ceci est d'ailleurs valide même si le graphe n'est pas biparti).

Considérons maintenant un couplage maximum M, de taille μ, et construisons une couverture de sommets de même taille. Notons $U \cup V$ la bipartition du graphe, et définissons Z_0 comme l'ensemble des sommets de U non-incidents à M, et Z comme l'ensemble des sommets accessibles Z_0 par des chemins alternants (une arête hors de M, puis une arête de M, etc).

On va montrer que $S = (U \setminus Z) \cup (V \cap Z)$ est une couverture de sommets de taille μ. Considérons une arête uv ($u \in U, v \in V$) qui n'est pas couverte, c'est-à-dire que ni u ni v n'appartient à S. Alors en particulier, $u \in Z$ et $v \notin Z$. Par définition, il existe un chemin alternant de Si $uv \notin M$, alors par définition, il existe un chemin alternant de Z_0 à u. Comme $u \in U$, la dernière arête de ce chemin appartient à M. En particulier, si $uv \in M$, v précède u dans ce chemin (et donc $v \in Z$, ce qui est impossible), et si $uv \notin M$, on peut prolonger ce chemin alternant jusqu'à v, ce qui assure encore que $v \in Z$. Ainsi, chaque arête est couverte.

Il faut maintenant montrer que $|S| = \mu$. Pour ce faire, on va montrer que chaque sommet de S est incident à une arête de M, et que chaque arête de M a exactement une extrémité dans S. Si $u \in U \cap S$, alors $u \notin Z$ donc $u \notin Z_0$, et en particulier u est incident à une arête de M. Si $v \in V \cap S$, alors v est l'extrémité d'un chemin alternant partant de Z_0, et ce chemin termine par une arête hors de M. Comme v n'est pas incident à une arête de M, on peut alors échanger les arêtes et non-arêtes de M de ce chemin et augmenter la taille de M, une contradiction. Ainsi, chaque sommet de S est incident à M.

Pour conclure, observons que chaque arête de M ne peut avoir qu'une extrémité dans S. En effet, si l'une de ses extrémités est dans Z, alors l'autre aussi. On a donc bien $|S| = |M| = \mu$. □

On peut maintenant utiliser ce résultat pour montrer le résultat de la partie.

Démonstration du Théorème 1. On rappelle (voir CM précédents et TD) que couplage maximum et couverture de sommets minimum peuvent être résolus par des programmes linéaires en nombres entiers, et surtout que ces programmes sont duaux l'un de l'autre. Ainsi, en notant μ la taille d'un couplage maximum, τ la taille d'une couverture minimale et μ^*, τ^* leurs variantes fractionnaires, le théorème de dualité forte donne

$$\mu \leqslant \mu^* = \tau^* \leqslant \tau.$$

D'après le théorème de König, $\mu = \tau$ et donc ces inégalités sont des égalités. En particulier, on peut calculer μ et τ avec la version fractionnaire des PLs correspondants, ce qui se fait donc en temps polynomial.

Enfin, pour calculer un indépendant maximum, il suffit de remarquer que le complémentaire d'une couverture de sommets minimum est un indépendant maximum. □

Remarque : si on ne souhaite pas utiliser de PL, on peut aussi retrouver μ et τ à l'aide d'un algorithme de flots.

1.2 Unit disk graphs

On se donne un ensemble d'antennes 5G réparties dans le plan, qui ont chacune une portée de 500m. Dans un premier temps, on souhaite trouver des clusters, c'est-à-dire des ensembles d'antennes dont les portées se chevauchent deux à deux, et qui soient maximaux.

Une modélisation naturelle de ce problème est de considérer le graphe dont les sommets correspondent aux antennes, et les arêtes relient des antennes dont les portées se chevauchent, c'est-à-dire à distance au plus 1km.

Théorème 3. *Dans un unit disk graph, on peut calculer une clique maximum en temps polynomial.*

Démonstration. Considérons deux points uv les plus éloignés dans une clique maximum. Le reste des sommets de la clique sont à distance au plus 1km de u et v, et appartiennent donc à l'intersection de deux disques de rayon 1km centrés en u et v. Observons la restriction du graphe aux sommets de ce domaine. Ceux situés au dessus du segment $[uv]$ forment une clique, tout comme ceux en dessous du segment. Ainsi, le complémentaire de ce graphe est biparti, et pour trouver une clique maximum dans le voisinage commun de u et v, il suffit de trouver un ensemble indépendant maximum dans ce graphe biparti, ce qu'on sait faire en temps polynomial.

On peut donc itérer cette recherche (construction du graphe biparti + calcul d'un indépendant maximum) sur toutes les arêtes uv du graphe, et conserver la plus grande clique obtenue. □

Un autre problème lié aux réseaux d'antennes est dû aux interférences : chaque antenne utilise des fréquences pour communiquer avec les appareils à sa portée, et si deux antennes utilisent la même fréquence, des interférences se produisent dans l'intersection de leur portées respectives. On peut alors se demander combien de fréquences doivent être utilisées au total pour pouvoir éviter les interférences. Une fois modélisé sous forme de graphes comme précédemment, le problème devient naturellement un problème de coloration.

Théorème 4. *Dans un unit disk graph, on peut calculer une 3-approximation d'une coloration optimale en temps polynomial.*

Démonstration. Considérons l'algorithme glouton consistant à colorer les sommets de gauche à droite en utilisant à chaque fois la plus petite couleur disponible. On va montrer que le nombre de couleurs utilisées est au plus $3\omega + 1$ où ω est la taille de la plus grande clique du graphe.

Soit v un sommet et supposons que tous les sommets à sa gauche sont colorés. On considère le demi-cercle de rayon 1km centré en v et à gauche de v, et on le découpe en trois secteurs de 60°. Chaque voisin coloré de v doit appartenir à un de ces secteurs, et observez (par Thalès) que la distance entre deux sommets d'un même secteur est d'au plus 1km. Ainsi, chacun de ces secteurs forme une clique, qui ne contient qu'au plus ω sommets.

Finalement, le voisinage de v ne peut contenir que 3ω sommets colorés au pire, et on pourra toujours colorer v avec une couleur plus petite que $3\omega + 1$.

En particulier, on a $\omega \leqslant \chi \leqslant 3\omega + 1$, l'algorithme construit donc une 3-approximation d'une coloration optimale. □

1.3 Graphes d'intervalles

Un problème classique de planification est le suivant : on se donne un ensemble de tâches, sous la forme d'une heure de début et une heure de fin. On dispose d'un ensemble de processeurs (ou de personnes) qui ne peuvent exécuter qu'une tâche à la fois. Ainsi, deux tâches dont les créneaux se chevauchent ne peuvent être effectuées par le même processeur.

La question principale vise à déterminer le nombre minimum de processeurs nécessaires pour mener à bien toutes les tâches.

Il s'agit encore une fois d'un problème de coloration d'un graphe bien choisi : les sommets correspondent aux tâches, et les arêtes relient des tâches dont les créneaux se chevauchent. Les couleurs correspondront ainsi aux processeurs à qui on assigne les tâches.

Théorème 5. *Dans un graphe d'intervalles, on peut calculer une coloration optimale en temps polynomial.*

Démonstration. On va colorer les sommets de manière gloutonne, en considérant les intervalles par ordre croissant de date de début, et en les colorant avec la plus petite couleur disponible.

Soit k la plus grande couleur utilisée par cet algorithme, et v un sommet coloré k. Considérons ce qu'il se passe au moment où v reçoit sa couleur. Alors, pour chaque couleur $i < k$, il doit exister un sommet v_i coloré i adjacent à v, et l'intervalle correspondant doit commencer avant v (sinon il ne serait pas encore coloré). Mais alors, tous les intervalles des v_i contiennent la date de début de v, et ils s'intersectent tous. On a donc trouvé un clique de taille k, et ainsi $k \leqslant \omega$.

On a donc réussi à colorer le graphe avec ω couleurs, ce qui est optimal (puisque le graphe contient une clique de taille ω par définition). □

D'un point de vue orthogonal, on peut aussi se demander la quantité maximale de tâches qu'on peut confier à un seul processeur. Il s'agit donc de trouver un ensemble indépendant maximum du graphe.

Théorème 6. *Dans un graphe d'intervalles, on peut calculer un ensemble indépendant maximum en temps polynomial.*

Démonstration. On considère encore un algorithme glouton. On prend le sommet dont l'intervalle se finit le plus tôt, on l'ajoute à l'indépendant qu'on calcule, puis on le supprime du graphe ainsi que ses voisins. On construit ainsi un ensemble indépendant maximal (par inclusion) I.

Montrons que I est en fait maximum. Parmi tous les ensembles indépendants maximum du graphe, notons I_m celui qui coïncide avec I le plus longtemps, c'est-à-dire qui maximise la date de fin de l'élément de $I \setminus I_m$ qui termine le plus tôt.

Si $I \not\subset I_m$, soit v le premier sommet de $I \setminus I_m$. Par définition, I et I_m coïncident avant v. Comme I_m est maximum, v doit être adjacent à un sommet w de I_m, qui termine donc plus tard que v.

Considérons l'ensemble J constitué de v, des éléments de I qui finissent avant v, et des éléments de I_m qui finissent strictement après w. Alors J est un ensemble indépendant de même taille que I_m, et qui coïncide avec I jusqu'après v, une contradiction avec le choix de I_m. On a donc $I \subset I_m$. Mais comme I est maximal par inclusion, on a $I = I_m$, donc I est maximum. □