Cours *conception et analyse d'algorithmes*
Correction du TD 5

1. $\mathcal{NP}$-complétude de HITTING-SET

Soit $(G = (V, E), k)$ une instance de RECOUVREMENT-PAR-SOMMETS (le graphe G est non orienté). Posons $S = X$, $C = E$. La question HITTING-SET(S, C, k) est vraie si et seulement s'il existe $S' \subset S$, $|S'| \leq k$, tel que $\forall x \in E, x = \{u, v\}$ on a $u \in S'$ ou $v \in S'$, ce qui est exactement RECOUVREMENT-PAR-SOMMETS(G, k). On a donc réduit RECOUVREMENT-PAR-SOMMETS à HITTING-SET. Comme ce dernier est manifestement dans $\mathcal{NP}$(la donnée de S' constitue un certificat de taille $|S'| \leq |S|$, donc polynomiale, qui se vérifie en énumérant les ensembles de C et en parcourant S' pour chaque élément, soit en temps $O(\text{taille}(C)|S'|)$), il est donc bien $\mathcal{NP}$-complet.

2. $\mathcal{NP}$-complétude de MIN-COUPE-CIRCUIT

i. Version de décision du problème : La donnée est un graphe orienté $G = (X, A)$ et un entier k il faut déterminer s'il existe un ensemble C composé de k arcs tel que $(X, A \setminus C)$ est sans circuit.

Ce problème est dans la classe NP car à partir d'un sous-ensemble C de A on calcule en temps polynomial une représentation du graphe $(X, A \setminus C)$ et on détermine par un parcours en profondeur si ce graphe est sans circuit.

ii. Soit C' l'ensemble obtenu à partir de C en remplaçant tous les arcs de la forme (x_i^1, x_j^0) par (x_i^0, x_i^1). Si $(X, A \backslash C')$ admet un circuit alors il passe néssairement par un arc de C qui n'est pas dans C' puisque $(X, A \setminus C)$ est sans circuit. Ce ne peut être qu'un arc de la forme (x_i^1, x_j^0), l'arc qui prècède (x_i^1, x_j^0) dans ce circuit est nécessairement (x_i^0, x_i^1), or par construction de C' il n'est pas dans $(X, A \setminus C')$; cette contradiction permet de prouver que $(X, A \setminus C')$ est bien sans circuit.

iii. On va montrer que le problème *RECOUVREMENT par SOMMETS* se réduit à la version de décision de *MIN-COUPE-CIRCUIT*. La donnée de *RECOUVREMENT par SOMMETS* est un graphe $G = (X, E)$ et un entier k, il s'agit alors de vérifier s'il existe un sous-ensemble Y de X ayant k éléments et tel que toute arête de E a au moins une extremité dans Y. Construisons le graphe $G' = (X', A)$ comme suggéré, il nous faut montrer que G admet un recouvrement par k sommets si et seulement si G' admet un coupe-circuit de taille k. Pour cela on vérifie que $Y = \{x_1, x_2, \ldots, x_k\}$ est un recouvrement par sommets de G si et seulement si $C = \{(x_1^0, x_1^1), (x_2^0, x_2^1), \ldots, (x_k^0, x_k^1)\}$ est un coupe-circuit de G' :

— Soit Y= $\{x_1, x_2, \ldots, x_k\}$ un recouvrement par sommets de G ; un circuit de G' passe par un arc (x^1, y^0) donc par (x^0, x^1) et (y^0, y^1) ; il correspond à une arête $e = \{x, y\}$ de G. Au moins une des extremités de e est dans Y, par exemple x, ainsi (x^0, x^1) est dans C. Tout circuit de G' passe par un arc de C, C est un coupe-circuit.

— Réciproquement si C est un coupe-circuit de taille k de G' on peut supposer qu'il est inclus dans A' par **2.**, ainsi $C = \{(x_1^0, x_1^1), (x_2^0, x_2^1), \ldots, (x_k^0, x_k^1)\}$. L'ensemble $Y = \{x_1, x_2, \ldots, x_k\}$ est un recouvrement par sommets de G car pour toute arête $\{x, y\}$ de G, (x^0, x^1), (x^1, y^0), (y^0, y^1), (y^1, x^0) est un circuit de G' qui doit passer par un arc de C qui est soit (x^0, x^1) soit (y^0, y^1), ainsi x ou y est alors dans Y.

3. Sous-ensemble acyclique maximal

On considère les 2 sous-ensembles formés pour l'un des arêtes croissantes (i, j) avec $i < j$ et pour l'autre des arêtes decroissantes. L'un de ces 2 contient au moins la moitié des arêtes et ils sont tout 2 acycliques.

4. $\mathcal{NP}$-complétude de RECOUVREMENT EXACT

i. Les $n(n-k)$ premières entrées de x doivent contenir exactement un 1 dans chacun des $n-k$ premirs groupes de n colonnes pour satisfaire les conditions sur les lignes 1 à $n-k$. Les coordonnées de ces premiers 1 doivent de plus différer modulo n pour éviter les collisions dans les n dernières lignes. Pour les k lignes restantes il faut mettre k 1 parmi les n dernières entrées de x. Inversement, pour tout choix de k positions parmi les n dernières on retrouve un vecteur x tel que $Bx = 1$ ayant des 1 à ces positions.

ii. Recouvrement exact est dans NP : on teste l'égalité $Ax = 1$ en temps quadratique.

Soit (G, k) une instance de Stable, posons $G = (X, E)$, $n = |X|$ et $m = |E|$. À G on associe la matrice C de taille $(2n-k+m) \times (n(n-k+1))$ formée par la matrice B de la question 1, à laquelle on ajoute m lignes contenant la matrice d'incidence arêtes/sommets dans le dernier blocs de n colonnes.

Tel quel si on associe à tout sous-ensemble S de k sommets le vecteur x_S tel que $Bx = 1$ qui marque les sommmets choisis par des 1 dans ses n dernières coordonnées, alors dans le produit Cx on a une ligne qui donne un coefficient 2 si une arête a ses 2 extrémités couvertes : S est un stable si et seulement si x_S n'a pas de coordonnées égales à 2.

Cependant il reste des coordonnées nulles dans Cx. On forme donc une matrice A de taille $(2n-k+m) \times (n(n-k+1)+m)$ en ajoutant m colonnes à C contenant des zéros et un bloc Id_m en regard des m lignes associées aux arêtes. À un vecteur x_S comme ci-dessus on associe alors le vecteur $\bar{x}_S$ de taille $2n-k+m$ avec des 1 aux positions associées aux colonnes de A correspondant aux lignes de Cx qui étaient nulles. Si S est un stable alors $A\bar{x}_S = 1$. Réciproquement tout vecteur $\bar{x}$ tel que $A\bar{x} = 1$ donne bien un x tel que Cx n'a de zéro que pour certaines lignes associées à des arêtes, et donc un stable. Ainsi G a un stable de taille exactement k si et seulement s'il existe une solution au problème RecouvrementExact pour cette matrice A.

5. Le problème du voyageur de commerce

i. Rappelons que l'arbre couvrant minimal est construit en triant les arêtes par poids croissant, et en ajoutant, dans l'ordre, les arêtes qui peuvent l'être sans créer un cycle.

Pour savoir si une arête crée un cycle avec les arêtes de la forêt déjà construite, il faut déterminer si ses deux extrémités sont dans le même arbre de cette forêt. On doit donc disposer d'une structure sur les sommets permettant de faire une requête rapide d'identification de l'arbre d'appartenance. En utilisant une structure Union-Find pour lequel une requête d'identification est en $O(\log(n))$ et l'union en temps $O(1)$, le test d'une arête coûte alors un temps total $O(\log n)$, soit un temps total pour la construction de l'arbre en $O(n^2 \log n)$, qui est du même ordre que le coût du tri des arêtes.

ii. Si l'on enlève une arête d'une tournée optimale, on obtient un arbre (puisqu'on enlève le cycle présent) couvrant, de coût inférieur au coût de la tournée, et supérieur au coût d'un arbre recouvrant minimal. Il s'ensuit que $c(H^*) \geq c(T)$.

iii. Le parcours complet revient à décrire exactement deux fois chacune des arêtes rencontrées dans le parcours en profondeur. Il s'ensuit facilement que $c(W) = 2c(T) \leq 2c(H^*)$.

Si l'on considère le parcours complet et que l'on supprime toutes les occurrences de chaque sommet, sauf la première, on obtient exactement le parcours en profondeur de l'arbre. Cette suppression revient à remplacer le chemin $a \to b \to c$ par $a \to c$, et d'après l'inégalité triangulaire, ne rallonge donc pas la tournée. Il s'ensuit que $c(H) \leq c(W) \leq 2c(H^*)$. On dispose donc d'un algorithme d'approximation à un facteur 2 pour le problème du voyageur de commerce avec inégalité triangulaire.

iv. Considérons le graphe à n sommets numérotés de 1 à n, et dont le poids des arêtes est donné par :

- $w(1, i) = 1$ pour tout $i \geq 2$;
- $w(i, i+1) = 2$ pour tout $2 \leq i \leq n-1$;
- $w(i, j) = 1 + \eta$ sinon.

L'arbre couvrant minimal est alors constitué des arêtes $(1, i)$ pour tout i. Le chemin obtenu consiste alors, si l'on suppose que le parcours en profondeur énumère les sous-arbres par racine d'étiquette croissante, à parcourir les sommets dans l'ordre de leur numéro ; le coût total est alors $1 + 2(n-2) + 1 = 2n - 2$.

Le parcours $1, 2, 4, 6, \ldots, 2\lfloor n/2 \rfloor 3, 5, 7, \ldots 2\lceil n/2 \rceil - 1, 1$ est quant à lui de longueur $2+(n-2)(1+\eta) = n+(n-2)\eta$. Le facteur d'approximation vaut donc $2-(2+2(n-2)\eta)/(n+(n-2)\eta)$, ce qui, en choisissant bien η, peut être rendu arbitrairement proche de 2.