Corrigé de la PC 5

19 février 2004

1. Le problème du voyageur de commerce en métrique

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.

Si l'on suppose que la recherche d'un cycle éventuel se fait en utilisant un Union-Find pour lequel la recherche est en $O(\log |A|)$ 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 seconde phase de l'algorithme en $O(n^2 \log n)$, qui est du même ordre que le coût du tri.

(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 $c(H) \leq c(W) \leq 2c(H^*)$.

(iv) Le résultat obtenu en **(iii)** montre que l'on dispose d'un algorithme d'approximation à un facteur 2 pour le problème du voyageur de commerce avec inégalité triangulaire.

(v) 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.

2. Couverture par ensembles

i) La preuve de l'exercice 2 de la PC4 montre que pour la version de décision, le cas $f = 2$ permet de réduire COUVRANT-MIN, donc est NP-complète ; la version générale l'est donc a fortiori (elle reste bien dans NP par le même argument que dans la PC4).

ii) On associe à chaque ensemble S une variable entière $X_S \geq 0$; l'idée est que X_S prendra la valeur 1 si l'on sélectionne l'ensemble S, la valeur 0 sinon. Le fait que chaque x_i soit dans au moins un des ensembles choisis fournit alors le système de contraintes

$$\sum_{S/x_i \in S} X_S \geq 1$$

tandis que la fonctionnelle à minimiser est $\sum_S X_S$.

iii) Posons Y_S l'arrondi de la solution réelle optimale X_S^*, comme défini dans l'énoncé. Alors, pour chaque x_i on a $\sum_{S/x_i \in S} X_S^* \geq 1$. Or, par hypothèse, la somme est une somme d'au plus f termes positifs ; l'un au moins de ces termes, disons $X_{S_0}^*$ est donc au moins égal à $1/f$; il s'ensuit que $Y_{S_0} = 1$, et donc que la collection d'ensembles construite en prenant les S tels que $Y_S = 1$ couvre bien tous les x_i, et est donc bien une couverture d'ensembles.

iv) La première partie est évidente : comme on a montré en **iii)** que la solution arrondie vérifie les contraintes du programme linéaire, elle est forcément moins bonne que la solution optimale du programme linéaire. À l'inverse, l'opération d'arrondi vérifie $Y_S \leq f X_S^*$ (noter que les X_S sont positifs), donc $\sum_S Y_S \leq f \sum_S X_S^* \leq f \cdot OPT$, ce qui montre que l'algorithme est un algorithme d'approximation à un facteur f.

v) Dans le cas $f = 2$, on considère l'instant de COUVRANT-MIN avec 2n sommets $A \cup B$, A est un stable de taille $.99n$, B est un graphe complet de taille $1.01n$, et on a le graphe biparti complet entre A et B. Le couvrant minimum a valeur $|B| = 1.01n$, le programme linéraire a pour solution optimale de donner poids $1/2$ a tous les sommets, de valeur n, et son arrondi a valeur $2n$.

3. Stable de taille maximum

i) Soit $Y \subset X$ un ensemble stable de taille k de G. Considérons le sous-ensemble $Y \times Y$ de $X \times X$. Si une arête relie deux sommets de $Y \times Y$ (u, u') et (v, v'), il s'ensuit qu'une arête de G relie soit u à v, soit u' à v'. Les deux entraînent que Y n'est pas un stable de G, impossible. Donc $Y \times Y$ est un stable de $G^{(2)}$ de cardinal k^2.

ii) Inversement, soit Z un stable de $G^{(2)}$ de cardinal m. Alors, pour tout couple $(u, u'), (v, v')$ d'éléments de Z, on a et $(u, v) \notin E$, et $u \neq v$ ou $(u', v') \notin E$. Considérons alors la partie Y de G formée des u tels que $(u, v) \in Z$ pour un certain v. La caractérisation précédente montre que (u, v) n'est pas une arête, et donc Y est bien un stable de G.

Pour u dans Y définissons $Y_u = \{v, (u, v) \in Z\}$. Montrons que Y_u est un stable de G : s'il existe une arête de G reliant v à v', il existe dans $G^{(2)}$ une arête qui relie (u, v) et (u, v'), qui appartiennent tous deux à Z, ce qui contredit le fait que Z soit stable. Donc Y_u est un stable.

Maintenant, on peut partitionner Z selon son premier élément, ce qui conduit à $m = \sum_{u \in Y} \#\{(u, v), v \in Y_u\} = \sum_{u \in Y} \#Y_u \leq \#Y \max_u \#Y_u$. Il s'ensuit que soit $\#Y \geq \sqrt{m}$, soit il existe un Y_u de cardinal au moins égal à $\sqrt{m}$. Dans les deux cas, l'assertion est prouvée.

iii) Supposons que l'on dispose d'un algorithme A d'approximation à un facteur c de MAXSTABLE, et construisons un nouvel algorithme B de la façon suivante : on fait tourner A sur $G^{(2)}$, et du stable retourné on construit comme en 3. un stable de G. Montrons que ce nouvel algorithme produit un résultat avec un facteur d'approximation $\sqrt{c}$.

Soit k la taille maximale d'un stable de $G^{(2)}$. La propriété de c-approximation de A montre que le stable renvoyé est de taille k' avec $k' \leq c \cdot k$.

La question i) montre que le stable maximum de G est alors de taille $\sqrt{k}$ – noter que k est un carré parfait – et B retourne dans le pire des cas un stable de taille $\sqrt{ck}$. On a donc $\sqrt{ck} \leq \sqrt{c}\sqrt{k}$. Par suite, A' est un algorithme d'approximation à un facteur $\sqrt{c}$ de STABLE.

iv. S'il existe un algorithme d'approximation à un facteur c, alors la question précédente montre que pour tout k il existe un algorithme d'approximation à un facteur $c^{1/2^k}$, qui peut être rendu arbitrairement proche de 1. Si l'algorithme initial était de complexité $O(f(n))$, l'algorithme obtenu est de complexité $O(f(n^{2^k}))$, qui reste polynomial à k fixé ; le schéma d'approximation construit n'est toutefois pas de type totalement polynomial (polynomial en n et en $1/k$).

L'exercice 4 sera traité lors de la PC6.