École polytechnique
Promotion 2004

Majeure 2 informatique

Cours de conception et analyse d'algorithmes

PC 5 – problèmes NP-complets et algorithmes d'approximation

14 février 2006

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

Soit $G = (X, A)$, un graphe orienté où X est l'ensemble des sommets et $A \subset X \times X$ celui des arcs (orientés). On rappelle qu'un circuit est une suite d'arcs de la forme :

$$(y_1, y_2), (y_2, y_3), \ldots, (y_{k-1}, y_k), (y_k, y_1)$$

Un *coupe-circuit* est un sous-ensemble C de A tel que le graphe $(X, A \backslash C)$, obtenu en supprimant les arcs de C est sans circuit.

On examine ici le problème MIN-COUPE-CIRCUIT de la détermination d'un coupe-circuit contenant un nombre d'arcs minimum.

i. Donner une version décision du problème et montrer que celle-ci est dans la classe $\mathcal{NP}$.

A partir d'un graphe non-orienté $G = (X, E)$ on construit un graphe orienté $G' = (X', A)$ de la façon suivante :

Soit $X = \{x_1, x_2, \ldots, x_n\}$, G' a $2n$ sommets :

$$X' = \{x_1^0, x_2^0, \ldots, x_n^0, x_1^1, x_2^1, \ldots, x_n^1\}$$

L'ensemble des arcs A est composé de deux sous-ensembles A' et A'' :

- $A' = \{1 \leq i \leq n | (x_i^0, x_i^1)\}$
- A partir de toute arête $\{x, y\}$ de E on construit 2 arcs (x^1, y^0) et (y^1, x^0) dans A''.

Ainsi le nombre d'arcs de G' est $2m + n$ où m est le nombre d'arêtes de G.

ii. Montrer que si C est un coupe-circuit de G', l'ensemble obtenu à partir de C en remplaçant tous les arcs (x_i^1, x_j^0) par (x_i^0, x_i^1) est encore un coupe-circuit. En déduire que l'on peut obtenir un coupe-circuit de G' ayant un nombre d'arcs inférieur ou égal à celui de C et ne contenant que des arcs de A'.

iii. Montrer que la version décision de MIN-COUPE-CIRCUIT est un problème $\mathcal{NP}$-complet ; vous utiliserez la construction proposée.

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

Etant donné un ensemble S de cardinal n et une collection C de sous-ensembles de S, et étant donné un entier k, le problème est de décider s'il existe un sous-ensemble S' de S de taille au plus k et qui contienne au moins un élément de chaque ensemble de C. Montrer, par réduction à partir de COUVRANT-MIN, que ce problème est $\mathcal{NP}$-complet.

3. $\mathcal{NP}$-complétude de l'isomorphisme de sous-graphes

Etant donné deux graphes G et H, le problème est de décider s'il existe une injection ϕ de V_H dans V_G telle que $\{u, v\} \in E_H$ ssi $\{\phi(u), \phi(v)\} \in E_G$. Montrer que ce problème est $\mathcal{NP}$-complet par réduction à partir de STABLE.

4. Sous-ensemble acyclique maximal.

Soit $G = (V, E)$ un graphe orienté, avec $V = \{1, 2, \ldots, n\}$. On souhaite choisir un sous-ensemble A des arcs, de cardinal maximum, tel que le sous-graphe (V, A) soit acyclique (c'est-à-dire qu'il ne contienne pas de circuit).

Présenter brièvement un algorithme qui, en temps polynomial, trouve un ensemble d'arcs acyclique de cardinal au moins OPT/2, où OPT est le cardinal d'un ensemble acyclique optimal. Le démontrer.

Indication : on pourra construire une partition de E en deux ensembles acycliques $E = E_1 \cup E_2$

5. Voyageur de commerce avec inégalité triangulaire

Ce problème consiste en la donnée d'un graphe complet non-orienté à n sommets numérotés de 1 à n (et donc $n(n-1)/2$ arêtes) ; à chaque arête est associée une distance. Le voyageur de commerce part du sommet 1 du graphe, et passe une fois et une seule par chacun des autres sommets du graphe avant de finalement revenir en 1, et ce en parcourant la distance minimale. La version "décision" de ce problème est un problème NP-complet.

On suppose que la distance vérifie l'inégalité triangulaire : pour tous sommets u, v et w, $d(u,v) \leqslant d(u,w) + d(w,v)$. On considère l'algorithme suivant :

- calculer un arbre couvrant minimal T enraciné en 1.
- renvoyer le tour H obtenu en parcourant les sommets dans l'ordre préfixe associé à l'arbre T.

i. Quelle est la complexité de cet algorithme ?

ii. Soit H^* une tournée optimale. Montrer que le poids $c(T)$ de T est inférieur ou égal à la longueur totale de H^*.

iii. En utilisant l'inégalité triangulaire, montrer que la longueur $c(H)$ de H est inférieure ou égale à $2c(T)$.

iv. En déduire un algorithme d'approximation à un facteur 2.

v. Montrer qu'il existe des cas où l'algorithme construit un tour de coût supérieur à $(2-\epsilon)c(H^*)$, et donc que l'analyse ne peut pas être améliorée.