Devoir : Magistère Informatique
Algorithmique de graphes
à rendre le jeudi 21 décembre 2006

Michel Habib
habib@liafa.jussieu.fr

17 janvier 2007

Parcours

1 : Ecrire un algorithme de calcul des composantes 2-arête connexes d'un graphe à l'aide d'un parcours en profondeur en s'inspirant de l'algorithme de Tarjan qui calcule les composantes fortement connexes.

Réponse : C'est exactement le même principe algorithmique que l'on peut adapter à la recherche des composantes 2-connexes ou 2-arête connexes d'un graphe.

A partir d'un parcours en profondeur sur un graphe non orienté, on peut orienter les arêtes dans le sens du parcours pour les arêtes qui permettent de marquer un nouveau sommet à explorer, les autres arêtes étant orientée dans le sens du "retour" (i.e. vers le sommet visité en premier).

Ce faisant il n'existe pas d'arc traversier, les arcs du coarbre sont soit de transitivité, soit ferment un circuit avec un chemin de l'arborescence de visite.

La même fonction calculée récursivement $RETOUR(x)$ pendant le parcours permet d'obtenir les composantes.

2 : Nous avons vu la résolution d'un système de clauses en logique des proposition à deux variables par clause (2-SAT), à l'aide de composantes fortement connexes d'un graphe associé au système de clauses. Comment adapter la transformation dans le cas où l'on considère Q-2-SAT : les quantificateurs sont autorisés ?

Complexité d'un algorithme de résolution.

Réponse : On construit le même graphe G et on calcule ses composantes fortement connexes et pour savoir s'il existe une solution, il suffit de vérifier sur ce graphe :

1) que pour chaque variable x, sa variable complémentaire n'appartient pas à composante fortement connexe de x.

2) Pour chaque paire de variables x, y qui sont reliés dans la formule quantitifiée par :

$\forall x, \dots \forall y$

il faut vérifier qu'il n'existe pas un chemin de x ou $\overline{x}$ vers y ou $\overline{y}$.

L'ensemble de ces vérifications nécessite $O(n+m)$.

Chemins

1 : Nom (*Chemin passant par K*)

Données (*Un graphe $G = (X, U)$ orienté dont les arcs sont valués par $\omega : U \longrightarrow Z$, K un entier et deux sommets s et p*)

Question (*G admet-il un chemin de s à p passant par K ?*)

Un chemin $\mu = [s = x_0, x_1, x_2, \dots, x_h = p]$ de G passe par K, s'il existe $i \in [1, h]$ tel que : $\Sigma_{j=0}^{j=i} \omega(x_j x_{j+1}) = K$

Montrer que ce problème est dans NP ?

Réponse :

Un problème de décision est dans NP, si lorsque la réponse est OUI, il existe un certificat de taille polynomiale vérifiable polynomialement.

Dans ce cas le chemin allant de s à p passant par K est le certificat polynomial.

2 : Montrer que ce problème de décision est NP-complet. Piste : on peut construire une réduction à partir de Somme Exacte.

Réponse : Rappelons la définition du problème Somme Exacte.

Nom (*Somme Exacte*)

Données (*n entiers $a_1, \dots a_n$ et un entier K*)

Question (*Existe-t-il $I \subseteq [1, n]$ tel que $\Sigma_{i \in I} a_i = K$?*)

Il existe plusieurs façons de faire, l'une des plus simple à décrire est la suivante :

À chaque instance de Somme Exacte, on associe un graphe constitué d'un chemin unique $[s = x_0, x_1, \dots x_{n-1}, x_n = p]$ que l'on complète en ajoutant tous les arcs de transitivité. Les valuations des arcs sont 0, sauf pour ceux du chemin et $\omega(x_{i-1} x_i) = a_i$ pour $i \in [1, n]$.

Il est facile de vérifier qu'il existe un chemin de s à p passant K ss'il existe une solution au problème de Somme Exacte.

3 : Que devient le problème si l'on impose que la donnée soit un graphe sans circuit ?

Réponse : Le problème reste NP-complet, car dans la réduction ci-dessus, le graphe utilisé est sans circuit.

4 : Etudions une variante du problème précédent :

Nom (*Chemin évitant K*)

Données (*Un graphe $G = (X, U)$ orienté dont les arcs sont valués par $\omega : U \longrightarrow Z$, K un entier et deux sommets s et p*)

Question (*G admet-il un chemin de s à p évitant K ?*)

Un chemin $\mu = [s = x_0, x_1, x_2, \ldots, x_h = p]$ de G évite K, si $\forall i \in [1, h]$ $\Sigma_{j=0}^{j=i} \omega(x_j x_{j+1}) \neq K$.

5 : A-t-on Chemin passant par K = co-(Chemin évitant K) ?

Réponse : Les deux problèmes ne sont pas complémentaires. Il suffit de considérer le graphe suivant possédant quatre sommets s, p, a, b, les arcs sa, ap, sb, bp dont les valuations sont :

$\omega(sa) = \omega(ap) = 1$ et $\omega(sb) = \omega(bp) = 2$.

Avec $K = 1$, il existe à la fois un chemin évitant 1, le chemin $[s, b, p]$ et un chemin passant par 1, le chemin $[s, a, p]$.

Sur cette instance les deux problèmes admettent la réponse OUI.

6* : Est-ce polynomial ou NP-complet ?

Dans le cas polynomial, on évaluera la complexité de l'algorithme de résolution.

Réponse : Le problème est polynomial, pour s'en convaincre il suffit de construire un algorithme s'inspirant du paradigme de la programmation dynamique. Il suffit d'explorer le graphe à partir du sommet s en gardant pour chaque sommet x une liste d'au plus n "premières" valuations différentes de chemins allant de s à x : $v_1(x), \ldots, v_k(x)$.

Au cours de l'exploration d'un sommet x du graphe :

Pour chaque arc xy de G,

tout valeur $v_i(x)$ telle que $v_i(x) + \omega(xy) \neq K$ est transmise à y dans la limite de n valeurs distinctes.

L'algorithme arrive à transmettre une valeur au sommet p ss'il existe un chemin évitant K dans G.

La complexité est polynomiale., mais cet algorithme peut surement être amélioré je conjecture en O(n+m) ?

Graphe fortement connexe minimal

Etant donné un graphe fortement connexe, il est possible d'enlever des arcs de transitivité tant que possible, le graphe ainsi obtenu est toujours fortement connexe mais sans arc de transitivité. Un tel graphe est appelé fortement connexe minimal.

1 : Montrer sur un exemple, qu'il n'y a pas unicité.

Réponse : Considérons un graphe complet symétrique sur n sommets.

Tout circuit sur ces n sommets est fortement connexe minimal (ayant n arcs), mais aussi toute arborescence symétrique (ayant 2n arcs).

En outre il est NP-difficile de calculer un graphe fortement connexe ayant le nombre minimum d'arcs.

2* : Ecrire un algorithme qui calcule à partir d'un graphe fortement connexe donné un graphe fortement connexe minimal. Complexité de l'algorithme proposé. Piste : on peut utiliser un parcours en profondeur.

Réponse : L'algorithme naïf qui consiste à enlever les arcs de transitivité tant qu'il en existe, est polynomial mais la complexité est de l'ordre de $O(nm^2)$.

Il existe des algorithmes en $O(nm)$, et il est conjecturé l'existence d'un algorithme en $O(n+m)$.

On suppose le graphe G fortement connexe Si l'on utilise l'algorithme de Tarjan pour la recherche des composantes fortement connexes. Ce parcours partitionne les arcs du graphe en : arcs de l'arborescence liée au parcours, arcs de transitivité de l'arborescence, arc de retour, arc traversiers.

Comme G ets fortement connexe les arcs traversiers sont des arcs de transitivité.

Il ne reste donc plus que les arcs de retour, mais en chaque sommet il suffit de garder l'arc de retour qui remonte le plus haut dans l'arborescence. En chaque sommet nous gardons seulement au plus un arc du coarbre. Le parcours en profondeur nous permet d'obtenir un graphe G' fortement connexe ayant au plus $2(n-1)$ arcs. Pour terminer il suffit de vérifier la transitivité de $n-1$ arcs, ce qui nécessite au plus une complexité de $O(n.m)$