Master Informatique Fondamentale (1ère année)
Algorithmes d'approximation — **TD4**

Après le centre, l'asymétrie

29 octobre 2004

Remarque préliminaire : Un graphe $G = (V, E)$ avec des arêtes de poids 1 et 2 est métrique.
En effet, $\forall x, y, z \in V, w(xz) \leq 2 = 1 + 1 \leq w(xy) + w(yz)$.

Exercice 1 : k-centre. Nous avons vu en cours une 2-approximation du problème du k-centre métrique. Nous complétons ici notre connaissance du problème par quelques résultats négatifs.

Question 1. *Trouver une famille d'instances critiques pour l'algorithme vu en cours dans le cas $k = 1$. (Indice : considérer un graphe dont toutes les arêtes sont de coût 2 sauf celles qui sont incidentes à un sommet particulier qui sont de coût 1).*

Réponse : On prend, pour $n \geq 2$, $V = \{x, v_1, \ldots, v_n\}$ et $E = \{xv_i\}_{1 \leq i \leq n}$. On pose $\forall e \in E, w(e) = 1$. $G = (V, E)$ est une étoile (métrique).
L'algorithme trie les arêtes par w croissant : par exemple $\forall 1 \leq i \leq n, e_i = xv_i$. $\forall 1 \leq i \leq n, G_i^2$ admet $x, v_{i+1}, \ldots, v_n$ comme stable maximal. Sa taille est $n - i + 1 > k = 1$ tant que $i < n$, donc l'algo peut continuer jusqu'à ce que $i = n$; G_n^2 étant une clique, ses sommets pris 1 à 1 forment des stables maximaux ; l'algo peut donc retourner $\{v_1\}$; le coût est alors 2, au lieu de 1 pour l'optimal, qui est $\{x\}$ (car $n \geq 2$). $\square$

Question 2. *Montrer, dans l'hypothèse où $P \neq NP$, que $\forall \epsilon > 0$ il n'existe pas de $(2 - \epsilon)$-approximation du problème du k-centre métrique. (Indice : on pourra utiliser le fait que le problème de trouver un dominant de poids minimum est NP-difficile ; on remarquera par ailleurs que tout graphe orienté complet avec des poids 1 ou 2 est métrique).*

Réponse : Supposons qu'il existe une $(2 - \epsilon)$-approximation au problème du k-centre métrique pour un $\epsilon > 0$.

Soit $G = (V, E)$ une instance du problème du dominant. On pose $G' = (V, E')$, G' complet, et $\forall e' \in E', w'(e') = \begin{cases} 1 & \text{si } e' \in E \\ 2 & \text{sinon} \end{cases}$. Par la remarque préliminaire, G' est métrique.

$X \subseteq V$ est un dominant de taille $\leq k$ de G ssi X est un k-centre de G' de coût 1 (par construction de G').

Donc G admet un dominant de taille $\leq k$ ssi G' admet un k-centre de coût 1
ssi l'algorithme d'approximation retourne un k-centre de G' de coût 1 (car le coût d'un k-centre de G' est 1 ou est supérieur ou égal à 2).

Une $(2 - \epsilon)$-approximation au problème du k-centre métrique permettrait donc de résoudre le problème du dominant en temps polynomial ; ce problème étant NP-complet, on aurait alors $P = NP$. $\square$

Question 3. *Montrer, dans l'hypothèse où $P \neq NP$, que pour toute fonction $\alpha(n)$ calculable en temps polynômial le problème du k-centre général n'admet pas de $\alpha(n)$-approximation.*

Réponse : On va réduire le problème du k-centre à celui du dominant
On suppose qu'on dispose d'une $\alpha(n)$-approximation A pour le prolème du k-centre général.
Soit $G = (V, E)$, on va chercher un dominant de taille $\leq k$ (c'est NP-difficile).
On pose $G' = (V, E')$, complet, avec $\forall e \in E'$, $w(e) = 1$ si $e \in E$, et $w(e) = \alpha(n) + 1$ sinon.

- Si A renvoie un k-centre de poids $\leq \alpha(n)$,il existe un dominant de taille k dans G.
- Sinon, le k-centre utilise au moins une arête de poids $\alpha(n) + 1$, il n'existe pas de k-centre de poids 1, et donc pas de dominant de taille k.

Si le k-centre était $\alpha(n)$-approximable, on pourrait résoudre le problème du dominant, qui est NP-difficile. $\square$

Exercice 2 : TSP asymétrique. On s'intéresse ici au problème du voyageur de commerce dans un graphe orienté : étant donné $G = (V, E)$ orienté et muni d'une fonction de coût positive sur les arcs, il s'agit de trouver un circuit de coût minimum qui passe par chaque sommet exactement une fois.

Question 4. *Montrer, dans l'hypothèse où $P \neq NP$, que pour toute fonction $\alpha(n)$ calculable en temps polynômial ce problème n'admet pas de $\alpha(n)$-approximation.*

Réponse : Dans le cours, on a vu que TSP (représentant de commerce) est inapproximable pour toute fonction $\alpha(n)$ calculable en temps polynômial, en partant du cycle Hamiltonien (NP-difficile). Ici, on va procéder de même pour le TSP asymétrique (voyageur de commerce) en partant du circuit Hamiltonien.

En entrée du TSP asymétrique, on a une clique orientée $G = (V, E)$, avec des poids sur les arêtes. On cherche un circuit qui passe par tous les sommets de G une fois et une seule, de longueur minimum.

Soit $\alpha(n)$, calculable en temps polynomial. On part d'une instance du Circuit Hamiltonien, $G = (V, E)$ orienté. On crée une instance du TSP asymétrique $G' = (V, E')$ comple, avec comme poids sur les arêtes : $\forall e \in E'$, $w(e) = 1$ si $e \in E$, et $w(e) = n \times \alpha(n)$ sinon.

Supposons qu'il existe A, une $\alpha(n)$-approximation pour le TSP asymétrique. Soit A' l'algorithme suivant pour le circuit Hamiltonien :

- A partir d'une instance G du cicrcuit Hamiltonien, construire G', l'instance correspondante du TSP asymétrique.
- Lancer A sur G'.
- Si $A(G') \leq n * \alpha(n)$, alors OUI, G admet un Circuit Hamiltonien.
- Si $A(G') \geq n * \alpha(n) + 1$, alors NON, G n'admet pas de Circuit Hamiltonien.

En effet, par construction, si $A(G') \leq n * \alpha(n)$, alors $A(G') = n$ (l'algorithme a choisi exactement les arêtes de poids 1). Donc G admet un circuit Hamiltonien.

Par contre, si $A(G') \geq n * \alpha(n) + 1$, comme A est une $\alpha(n)$-approximation, $A(G') \leq \alpha(n) * TSP_{opt}(G')$. D'où : $TSP_{opt}(G') \geq n + 1/\alpha(n) > n$. G n'admet donc pas de circuit hamiltonien. □

On s'intéresse donc à présent à la version métrique du problème : tous les graphes considérés sont donc orientés complets et munis d'une fonction de coût vérifiant l'inégalité triangulaire (orientée).

Question 5. *Montrer que dans un graphe complet on peut calculer en temps polynômial une couverture des sommets par une famille de circuits de poids total minimum. (Indice : une couverture par circuits de $G = (V, E)$ est une permutation de V, et une permutation de V est un couplage parfait dans un certain graphe biparti complet).*

Réponse : On part du graphe initial $G = (V, E)$. On construit $G' = (V_1 \cup V_2, E')$ où V_1 et V_2 sont deux copies de V (1), et $\forall ab \in E, a_1b_2 \in E'$ (orienté). On obtient alors un graphe biparti sur lequel on va effectuer un couplage parfait, qui correspondra à une couverture par circuits de G :

En effet, le couplage parfait associe à chaque élément de V, un unique antécédent, et un unique successeur. ('couplage' : injectif, 'parfait' : sujectif). Le couplage est possible car on est dans un graphe complet (ceci est assuré par le fait qu'on étudie la version métrique du problème). Remarque : pour les questions suivantes, on aura besoin d'éliminer les boucles sur les singletons ; ceci n'empêche pas d'effectuer un couplage car le théorème de Hall s'applique toujours.

De même une couverture par circuits de V équivaut à associer à chaque élément de V un unique antécédent et un unique successeur.

La différence entre les instances utilisées pour chacun des deux problèmes considérés, est que pour le graphe biparti, $Successeur(V_1) \in V_2$, $Antcdent(V_2) \in V_1$. Tandis que pour la couverture par circuits, c'est l'orientation des arêtes qui définit les relations 'Successeur' et 'Antécédent'.

Rq : l'intermédiaire consistant à utiliser une permutation est beaucoup plus intuitif, surtout au niveau de la correspondance "permutation - cycle", mais n'est pas nécessaire pour répondre à cette question. En effet les notions importantes que sont l'injectivité et la surjectivité de la fonction associant un élément de V à un autre sont déjà présentes dans le couplage et dans le couverture par circuit de G.

La construction du graphe utilisé pour le couplage s'effectue en temps polynomial, puisqu'il s'agit de recopier V, et de construire E' qui a la même taille que E. Un couplage de graphe biparti s'effectue en temps polynomial, par exemple avec la méthode des chaînes augmentantes.

Enfin, remarquons que la fonction de poids est inchangée, par construction. □

On cherche à approximer la version métrique du problème selon le principe suivant : on cherche une couverture par cycles de poids minimum, on fusionne chaque cycle en un unique sommet puis on itère le processus.

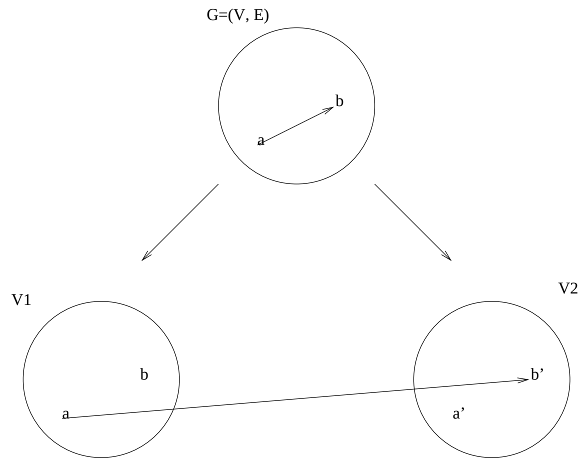

FIG. 1 – On réalise deux copies du graphe de départ

Question 6. *Écrire un algorithme polynomial fonctionnant selon ce principe et qui renvoie un cycle passant une fois et une seule par chaque sommet.*

Réponse :

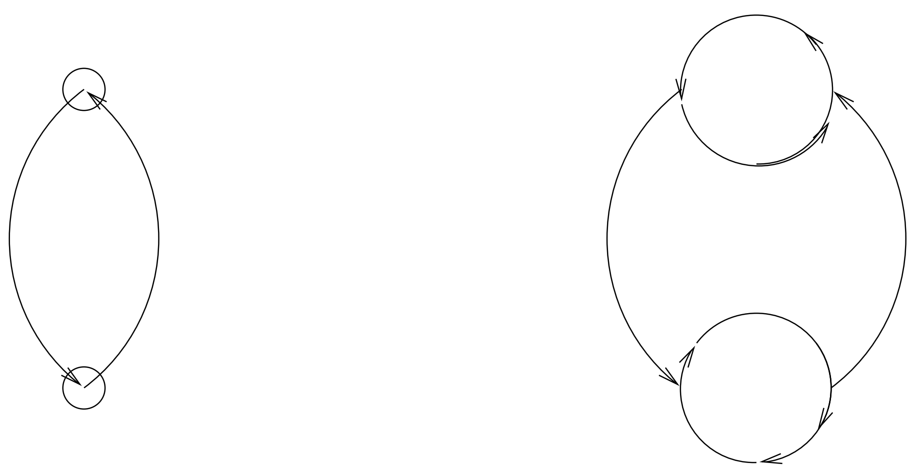

FIG. 2 – Il faut rééetendre le graphe...

Tout d'abord, quelques remarques sur l'algorithme :

- Lorsque l'on fusionne plusieurs cycle en sommets, on peut avoir plusieurs arêtes de poids différents qui joignent deux sommets ; on peut alors garder l'arête de poids minimal, sans perte.
- Après avoir fusionné tous les cycles en sommets uniques, il va falloir rééetendre les circuits tout en gardant un tour. Pour plus de simplicité (il existe plusieurs méthodes), on va procéder localement.

Sous-Algorithme 'FusionGraphe(Graphe G)' :

- Entrée : $G = (V, E)$.
- Dupliquer V en V_1 et V_2.
- Construire E' : $\forall ab \in E, a_1 b_2 \in E'$
- Effectuer un couplage sur (V_1, V_2, E').
- Construire la famille induite de cycles C, qui couvre G.
- Fusionner les cycles de C :
  - Chaque cycle devient un unique point
  - Entre deux points nouvellement construits, on garde uniquement une des arêtes qui existaient avant (poids minimum)

FIG. 3 – ...tout en gardant un tour.

- Nommer les nouveaux points, et stocker le cycle qui leur étaient associés. (On l'utilisera plus tard) .
- Renvoyer le nouveau graphe fusionné, et les correspondances, (Noeud 'nom') -> (Cycle 'cycle associé').

Sous-Algoritme 'Etendre(Noeud n)'
- Renvoyer le cycle associé au nom n.

Sous-Algorithme 'Tour(Cycle C, Noeud entree, Noeud sortie)'
- (cf 3)
- On suppose que les n noeuds du cycle sont numértoés, et que n(entree) = 0, n(sortie) = s ;
- retourner [0..s-1] : :[s+1..n-1] : :s

Algorithme 'Circuit(G)'
- (*On notera $G = (V, E)$ à chaque étape*)
- Tant que $|V| > 1$ faire
  - FusionGraphe(G)
- Fin Tant que
- (*A partir de ce point, on sait que G est toujours un cycle*)
- Tant que (il existe dans V des noeuds fusionnés)
  - Choisir un des noeuds fusionnés.
  - Soit a sont antécédent, n le noeud et s son successeur.
  - Remplacer n par $Tour(Etendre(n), a, s)$
- Fin Tant que
- (*On peut alors renvoyer $G = (V, E)$, avec V les noeuds originaux, et E le circuit associé*)

□

Question 7. *Montrer que l'algorithme est une $O(\log(n))$-approximation du problème.*

Réponse : A chaque itération, le nombre de sommet diminue de moitié. Il y a $\log(n)$ étapes.
Dans le graphe compressé, à chaque étape, la couverture par circuit que l'on obtient est plus petite que le vrai tour optimal dans le graphe de départ : $\forall i, p(C_i) \leq OPT$, où C_i est la couverture par cycle à l'étape i. Soit T_i le Tour obtenu à l'étape i. $p(T_{i-1}) \leq 2 * p(C_{i-1}) + p(T_i)$ (Le '2' vient de l'approximation, et on peut majorer $p(C_{i-1})$ par OPT). On a donc : $p(T_{i-1}) \leq 2 * OPT + p(T_i)$, donc $p(T_0) = O(\log(n)) * OPT$. □

Exercice 3 : Steiner asymétrique. Étant donné un graphe orienté $G = (V, E)$ muni d'une fonction de coût positive sur les arcs et dans lequel certains sommets sont étiquetés "requis" et un sommet est étiqueté "racine", le problème est de trouver un arbre (orienté) de poids minimum enraciné au sommet "racine" et qui atteint tous les sommets "requis".

Question 8. *Donner une réduction isofacteur du problème de la couverture par ensembles au problème de l'arbre de Steiner asymétrique.*

Réponse : Considérons une instance du problème de la couverture par ensembles : on a un univers $U = \{u_1, \ldots, u_n\}$, un ensemble de parties de U $S = \{s_1, \ldots, s_k\}$ avec une fonction de poids w sur ces parties. On construit une instance du problème de Steiner asymétrique par la réduction ϕ suivante :
- L'ensemble des sommets du graphe est constitué d'un sommet "racine", de k sommets $v_1 \ldots v_k$, et de n sommets $u_1 \ldots u_n$.
- On a des arcs e_i entre la racine et v_i, de poids $w(s_i)$.
- On a des arcs $f_{i,j} = (v_i, u_j)$ ssi $u_j \in s_i$, de poids nul.
- Les sommets requis sont les u_i.

Montrons que ϕ est une réduction isofacteur :
- Pour toute instance I de couverture par ensembles, $\phi(I)$ est une instance de Steiner asymétrique par la réduction.

- La réduction et son inverse se construisent bien en temps polynômial.
- Considérons une solution $\mathcal{S} = \{s_{i_1} \dots s_{i_p}\}$ d'une instance I de couverture par ensembles. L'image $\phi(\mathcal{S}) = \{e_{i_1} \dots e_{i_p}\} \cup \{f_{i_l,j}, l \in 1 \dots p, u_j \in s_{i_l}\}$ est bien une solution de $\phi(I)$. En effet, comme $\mathcal{S}$ est une couverture par ensembles, pour tout u_i de l'univers U, il existe un ensemble $s_{i_l} \in \mathcal{S}$ tel que $u_i \in s_{i_l}$. Donc pour tout sommet requis u_i, il existe un chemin $e_{i_l}, f_{i_l,i}$ reliant ce sommet à la racine. Enfin $w(\phi(\mathcal{S})) = \sum w(s_{i_l})$ (par construction de ϕ), donc $w(\phi(\mathcal{S})) = w(\mathcal{S})$.
- La réciproque se montre de la même manière.

□

Question 9. *Qu'en déduire quant-aux possibilités d'approximer le problème de l'arbre de Steiner asymétrique.*

Réponse : Si on avait une $\alpha(n)$-approximation du problème de Steiner asymétrique avec $\alpha(n) \in o(H_n)$, alors on aurait une $\alpha(n)$-approximation pour le problème de la couverture par ensembles, ce qui n'est pas possible[1]. On ne peut donc pas avoir asymptotiquement mieux qu'une H_n-approximation pour le problème de Steiner asymétrique. □

★

[1] En fait, on sait que la borne H_n est optimale en un sens plus fort...