Complexité et Graphe **2014-2015**
ENSTA

Algorithme d'approximation

Le problème du k-centre

Dans cette partie, nous allons nous concentrer sur le problème du k-centre : étant donné un ensemble de villes dont les distances sont spécifiées, choisir k villes afin d'installer des entrepôts de façon à minimiser la distance maximale d'une ville à l'entrepôt le plus proche. Un tel ensemble de k villes est appelé k-centre.

Nous allons nous focaliser sur le problème de k-CENTRE MÉTRIQUE. Ici, la fonction de poids que l'on notera w respecte l'inégalité triangulaire, c'est-à-dire pour tout triplet de villes u, v, et z, on a $w(u, v) \leq w(u, z) + w(z, v)$.

Exercice 1 Approximation du problème

Question 1.1 Le *carré* d'un graphe G, noté G^2, est défini comme le graphe qui contient une arête (u, v) pour chaque couple de sommets u et v reliés par un chemin de longueur au plus 2 dans G.

Montrer que si le sous-ensemble de sommets I est un stable de G^2 et alors pour tout ensemble dominant D de G, on a $|I| \leq |D|$.

Correction

Soit D^* un dominant minimum de G. Par définition, G possède $|D^*|$ étoiles couvrant tous les sommets de G. Comme les sommets d'une étoile de G forment une clique dans G^2,G^2 se décompose en au plus $|D^*|$ cliques qui couvent aussi tous ses sommets.

Par conséquent, tout stable I de G^2 ne pourra avoir au plus qu'un unique sommet dans chacune de ces cliques de G^2.

$$\text{Donc } |I| \leq |D^*|.$$

□

Question 1.2 Dans la suite de l'exercice, nous supposerons que les arêtes de K sont triées par poids croissant, c'est-à-dire $w(e_1) \leq w(e_2) \leq \cdots \leq w(e_m)$ et nous notons $K_i = (V, E_i)$, où $E_i = \{e_1, e_2, \ldots, e_i\}$.

Montrer que si G_i admet un dominant de taille au plus k alors K admet un k-centre de coût au plus $w(e_i)$

Question 1.3 Voici l'algorithme $\mathcal{A}$ du k-centre :

1. Construire $(G_1)^2, (G_2)^2, \ldots, (G_m)^2$.
2. Calculer un stable maximal, M_i, pour chaque $(G_i)^2$.
3. Calculer le plus petit indice i tel que $|M_i| \leq k$. Notons-le j.
4. Renvoyer M_j.

Donner la complexité de cet algorithme.

Notons OPT le coût d'un k-centre optimal. Notons aussi par i^* le plus petit indice, $w(e_i*) = OPT$ et tel que G_i* contient un k-centre optimal.

— Montrer que si j est l'indice calculé par l'algorithme, alors $w(e_j) \leq OPT$.
— Montrer que l'algorithme $\mathcal{A}$ est une 2-approximation.

Question 1.4 Considérons l'instance suivante : le graphe complet de $n+1$ sommets possède un sommet singulier u. Chaque arête incidente à u est de poids 1 et les autres sont de poids 2. Appliquer l'algorithme sur ce graphe avec $k = 1$ en considérant la pire solution. Qu'en déduisez-vous ?

Exercice 2 Problème Sac à Dos.

Le problème du Sac à Dos est un problème classique en informatique. Il modélise une situation analogue au remplissage d'un sac. Une personne veut remplir un sac à dos ne pouvant pas supporter plus d'un certain poids $C \in \mathbb{N}$, et elle dispose de n objets (On note l'ensemble des objects par $\mathcal{O} = \{1, \ldots, n\}$). Chaque objet i a une valeur $v_i \in \mathbb{N}\backslash\{0\}$ et un poids $p_i \in \mathbb{N}\backslash\{0\}$. Le problème est de trouver un ensemble d'objets tels que

— tous les objets de cet ensemble puissent être mis dans le sac.
— la somme des valeurs de ces objets soit maximale.

Le problème d'optimisation correspond à trouver un sous-ensemble d'objets dont le poids total est inférieure à C et dont la valeur totale soit maximum. **Notons $\mathbf{v} = \mathbf{max}\{\mathbf{v_i} : \mathbf{i} \in \mathcal{O}\}$.**

Rappelons l'algorithme de la programmation dynamique. Notons $T[i, j]$ représentera la valeur maximale pour un sac à dos de capacité j à l'aide des i premiers objets.

Entrée : un ensemble d'objects $\mathcal{O} = \{1, \ldots, n\}$. L'objet i a une valeur v_i et un poids p_i.
Sortie : un entier

1. Initialiser tous les éléments du tableau T à zéro.
2. Pour tout i allant de 1 à n faire
   (a) Pour tout j allant de 1 à C faire
   Si $j < v_i$, alors $T[i, j] = T[i-1, j]$
   sinon $T[i, j] = max(T[i-1, j], T[i-1, j-p_i] + v_i)$
3. Retourner $T[n, C]$

Question 2.1 Donner la complexité de cet algorithme. Peut-on modifier l'algorithme de telle façon que sa complexité soit exprimée en fonction de v ?

Maintenant, considérons l'algorithme suivant :

1. Etant donné $\varepsilon > 0$, $K \leftarrow \frac{\varepsilon \cdot v}{n}$
2. Pour chaque objet i, $v'_i = \lfloor \frac{v_i}{K} \rfloor$
3. Lancer l'algorithme de programmation dynamique avec ces valeurs arrondies v' et trouver un ensemble S' dont la valeur (v') totale soit maximum.
4. Renvoyer S'.

Question 2.2 Donner la complexité de cet algorithme.

Notons OPT le coût de la solution optimale.

Question 2.3 Montrer que $\sum_{i \in S'} v_i \geq (1 - \varepsilon) \cdot OPT$

Question 2.4 Qu'en concluez-vous ?

Exercice 3 Le problème du coloriage de graphe planaires

Un graphe est *planaire* si on peut le dessiner dans le plan sans que les arêtes ne se croisent. Soit $G = (V, E)$ un graphe planaire non vide *connexe*. Notons $n = |V|$, $a = |E|$.

Considérons un dessin de G dans le plan (sans que les arêtes ne se croisent). Soit f le nombre de régions du plan délimitées par les arêtes de G.
Question 3.1 Montrer que les graphes planaires respectent la relation d'Euler $n - a + f = 2$. (*Indication : raisonner par récurrence sur le nombre de faces*)

Question 3.2 On suppose $n \geqslant 3$. Montrer que $3f \leq 2a$.

Question 3.3 Montrer qu'il existe un sommet de degré au plus 5 dans le graphe planaire G.

Question 3.4 Concevoir un algorithme de 6-coloration en temps polynomial.

Remarque : tout graphe planaire peut être colorer avec 4 couleurs. Ce résultat est connu sous le nom de théorème des quatre couleurs. Il a été démontré en 1976 par Appel et Haken.

Exercice 4 Le problème de l'ensemble indépendant dans les graphes planaires

Le problème (de décision) de l'ensemble indépendant reste NP-complet même si les graphes sont planaires. Nous allons concevoir un algorithme calculant un ensemble indépendant de taille k pour les graphes planaires de n sommets en temps $O(6^k n)$.

Soit G un graphe planaire et k un entier. Par la question **3.3**, il existe un sommet u_0 un sommet de degré au plus 5 : on notera ces voisins $u_1, \ldots, u_d$ avec $d \leq 5$.

Question 4.1 Supposons que G possède un ensemble indépendant J de taille $k > 0$. Prouver que

1. Si J contient aucun sommet u_i avec $0 \leq i \leq d$, alors, il existe un ensemble indépendant de taille k contentant u_0.

2. Si J contient un sommet u_i avec $0 \leq i \leq d$, alors $J \backslash \{u_i\}$ est un ensemble indépendant de taille $k-1$ dans le graphe G_i, correspondant à G privé de tous les sommets v voisins de u_i et de leurs ses arêtes incidentes.

Question 4.2 Supposons que G ne possède pas un ensemble indépendant J de taille $k > 1$. Soit i entier entre 0 et d, Prouver que pour le graphe G_i correspondant à G privé de tous les sommets v voisins de u_i, ne possède pas un ensemble indépendant J de taille $k - 1$.

Considérons l'algorithme $\mathcal{A}(G, k)$ suivant

Entrée : un graphe planaire G et un entier k
Sortie : Vrai si et seulement si G possède un ensemble indépendant de taille k.

1. Si $k > |V(G)|$, alors renvoyer Faux.
2. Si $E(G) = \emptyset$ ou si $k = 0$, alors renvoyer Vrai.
3. Choisir un sommet u_0 de degré $d \leq 5$ avec pour voisins $u_1, \ldots, u_d$
4. Pour tout i allant de 0 à d, construire le graphe G_i tel que $V(G_i) = V(G) \backslash \Gamma_G(u_i)$ et $E(G_i) = E(G) \backslash \{e \in E : e \text{ a une extrémité dans } \Gamma_G(u_i)\}$
5. Renvoyer $\bigvee_{i=0}^{d} \mathcal{A}(G_i, k-1)$

Question 4.3 Exécuter cet algorithme sur une étoile à $n + 1$ sommets avec $k = 7$.

Question 4.4 Montrer que cet algorithme est correct.

Question 4.5 Donner la complexité de l'algorithme.

Question 4.6 Etendre ces résultats aux graphes t-dégénérés. Un graphe $G = (V, E)$ est t-*dégénéré* s'il existe un sommet v de degré inférieur ou égal à t dans G et si le graphe G privé de v possède la même propriété. Le graphe G privé de v est identique à G, sauf qu'on lui a retiré le sommet v ainsi que toutes les arêtes incidente à v.