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 $w(e_i)$

Correction

Si D est un dominant de taille au plus k dans G_i, alors tous les sommets de K sont à distance 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.

Correction

Construire la suite de graphes peut se faire en $O(mn)$ opérations.

Calculer un stable maximal peut se calculer en $O(n)$ opérations. Il suffit tout simplement parcourir tous les sommets et construire en parallèle un stable en ajoutant ou pas le noeud courant.

On calcule au pire des cas m stables. Cela nécessite $O(mn)$ opérations.

□

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 $w(e_j) \leq OPT$.

Correction

Pour tout entier $i < j$, on a $|M_i| > k$ (sinon l'algorithme retournerait i). D'après la question précédente, en notant D_i^* un dominant optimal du graphe G_i, $|D_i^*| > k$, et donc que $i^* > i$. Par conséquent, $j \leq i^*$ et $w(e_j) \leq w(e_{i^*})$. □

— Montrer que l'algorithme $\mathcal{A}$ est une 2-approximation.

Correction

Un stable maximal I d'un graphe est également un dominant : s'il existe un sommet v non dominé par I, alors $I \cup \{v\}$ serait un stable et ceci est en contradiction avec la propriété que I soit maximal.

Dans G_j^2, les sommets de M_j sont des centres des étoiles qui couvrent tous les sommets. Or, l'inégalité triangulaire donne que toutes les arêtes de G_j utilisées par ces étoiles ont un coût à $2w(e_j)$.

Comme $w(e_j) \leq w(e_{i^*})$, on retourne un k-centre de coût $2w(e_j)$ et $2w(e_j) \leq 2w(e_{i^*})$

□

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 ?

Correction

Pour $k = 1$, la solution optimale est le centre de la roue et $OPT = 1$.

La solution retournée par l'algorithme est la suivante $j = n$. En effet G_n^2 forme une clique. De plus si un sommet (autre que u) est sélectionné pour construire un stable maximal, alors le coût de la solution sera de 2.

□

Exercice 2 La coupe maximum d'un graphe

Nous allons considérer un graphe non-orienté $G = (V, E)$ ayant une fonction de poids sur les arêtes $w : E \to \mathbb{N}$. Une coupe d'un graphe non-orienté $G = (V, E)$ est un ensemble d'arêtes qui partagent G en deux sous-ensembles disjoints et distincts (S et $V \backslash S$). Dans la suite de l'exercice, la coupe se définit par aussi les deux sous-ensembles de sommets disjoints.

Cet exercice se traite le problème d'optimisation COUPE MAX : nous avons en entrée un graphe non orienté $G = (V, E)$ avec un poids $w(e)$ sur chaque arête e, et nous voulons partager les sommets en deux ensembles S et $V \backslash S$ afin que le poids total des arêtes entre les deux ensembles est aussi grand que possible.

Notation : Soit A et B deux ensembles de sommets disjoints. Nous noterons

$$w(A, B) = \sum_{u \in A, v \in B, (u,v) \in E} w(u, v) \text{ et } W(A) = \sum_{u \in A, v \in A, (u,v) \in E} w(u, v) \tag{1}$$

Rappellons que $\Gamma_G(u)$ est l'ensemble des sommets de G adjacents à u. Nous dirons qu'un sommet v est *α-content* pour la coupe $(A,\ B)$ dans le graphe G si et seulement si

— $w(A \backslash \{u\}, B \cup \{u\}) \leq \alpha w(A, B)$ si $u \in A$

— ou $w(A \cup \{u\}, B \backslash \{u\}) \leq \alpha w(A, B)$ si $u \in B$

Considérons le graphe biparti complet de 8 sommets (voir la figure 1). Toutes les arêtes de ce graphe ont un poids égal à 1 : i.e. $\forall e \in E,\ w(e) = 1$.

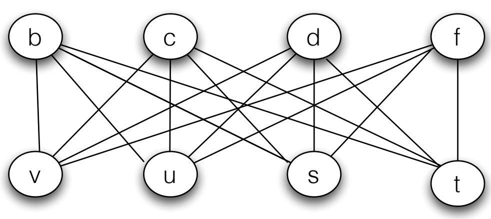


FIGURE 1 – Graphe biparti. Toutes ses arêtes ont un poids égal à 1.

Question 2.1 Compléter les trois tableaux

A	$B = V \backslash A$	$w(A, V \backslash A)$
$\{b\}$	$\{c, d, f, v, u, s, t\}$	4
$\{b, v\}$		
$\{b, c, v\}$		
$\{b, c, d, f\}$		

A	B	$A \backslash \{b\}$	$w(A \backslash \{b\}, B \cup \{b\})$	$w(A, B)$
$\{b, c, d, f\}$	$V \backslash A$	$\{c, d, f\}$	12	16
$\{b, c, d, v\}$	$V \backslash A$			
$\{b, u, s, t\}$	$V \backslash A$			

A	B	$W(A)$	$W(B)$	b est 1-content
$\{b, c, d, f\}$	$V \backslash A$	0	0	oui car ...
$\{b, c, d, v\}$	$V \backslash A$			
$\{b, u, s, t\}$	$V \backslash A$			

Correction

A	$B = V \backslash A$	$w(A, V \backslash A)$
$\{b\}$	$\{c, d, f, v, u, s, t\}$	4
$\{b, v\}$	$\{c, d, f, u, s, t\}$	6
$\{b, c, v\}$	$\{d, f, u, s, t\}$	2+3+3
$\{b, c, u\}$	$\{d, f, v, s, t\}$	8
$\{b, v, u\}$	$\{c, d, f, s, t\}$	8
$\{b, c, v, u\}$	$\{d, f, s, t\}$	8
$\{b, c, d, f\}$	$\{v, u, s, t\}$	16

A	B	$A\backslash\{b\}$	$w(A\backslash\{b\}, B\cup\{b\})$	$w(A,B)$
$\{b,c,d,f\}$	$V\backslash A$	$\{c,d,f\}$	12	16
$\{b,c,d,v\}$	$V\backslash A$	$\{c,d,v\}$	8	10
$\{b,c,u,v\}$	$V\backslash A$	$\{c,u,v\}$	8	8
$\{b,u,s,t\}$	$V\backslash A$	$\{u,s,t\}$	12	10

A	B	$W(A)$	$W(B)$	b est 1-content
$\{b,c,d,f\}$	$V\backslash A$	0	0	oui car $w(A\backslash\{b\}, B\cup\{b\}) \le \alpha w(A,B)$
$\{b,c,d,v\}$	$V\backslash A$	3	3	oui car $w(A\backslash\{b\}, B\cup\{b\}) \le \alpha w(A,B)$
$\{b,c,u,v\}$	$V\backslash A$	4	4	oui car $w(A\backslash\{b\}, B\cup\{b\}) \le \alpha w(A,B)$
$\{b,u,s,t\}$	$V\backslash A$	3	3	non car $w(A\backslash\{b\}, B\cup\{b\}) > \alpha w(A,B)$

□

Le problème COUPE MAX est trouver une coupe (A, B) qui maximise $w(A, B)$ sur l'ensemble des sous-ensembles de V. Considérons l'algorithme suivant pour le COUPE MAX :

Algorithme Max-Cut-Local

Entrée : Un graphe $G = (V, E)$ et une fonction poids $w : E \to \mathbb{N}$

Sortie : Deux sous-ensembles de sommets A et B.

1. Choisir une partition arbitraire de sommets (A, B) de V.
2. Tant qu'il existe un sommet v qui n'est pas α-*content*
   — Si v est dans A, alors $A \leftarrow A\backslash\{v\}$ et $B \leftarrow B \cup \{v\}$
   sinon $B \leftarrow B\backslash\{v\}$ et $A \leftarrow A \cup \{v\}$
3. retourner les deux ensembles (A, B)

Question 2.2 Exécuter l'algorithme ayant en entrée le graphe de la figure 1 sachant que $\alpha = 1$ et que la partition initiale est la suivante

1. $A = \{b, c, d\}$ et $B = \{v, u, s, t, f\}$
2. $A = \{v, u, b, c\}$ et $B = \{d, f, s, t\}$

Correction

Pour l'exécution sur la partition initiale $A = \{v, u, b, c\}$ et $B = \{d, f, s, t\}$: Tous les sommets sont contents. L'algorithme retourne cette partition.
Pour l'exécution sur la partition initiale $A = \{b, c, d\}$ et $B = \{v, u, s, t, f\}$: seul le sommet f n'est pas content. L'algorithme modifie la partition. Elle devient $A = \{b, c, d, f\}$ et $B = \{v, u, s, t\}$. Ici, dans la nouvelle partition, tous les sommets sont contents. L'algorithme retourne cette partition.

□

Par la suite, la solution retournée par l'algorithme Max-Cut-Local est notée (A, B) tandis une partition correspondant à une coupe optimale est notée (A^*, B^*).

Question 2.3 Donner une relation entre $W(E)$, $W(A)$, $W(B)$, $w(A, B)$.

Correction

$$W(E) = W(A) + W(B) + w(A, B).$$

□

Question 2.4 Montrer que pour tout sommet $u \in A$, on a $w(\{u\}, A) \leq w(\{u\}, B) + (\alpha - 1)w(A, B)$. *Indication : u est α-content.*

Correction

Par définition on a $w(A \backslash \{u\}, B \cup \{u\}) = w(A, B) + w(\{u\}, A) - w(\{u\}, B)$. Comme u est α-content, on a $w(A \backslash \{u\}, B \cup \{u\}) \leq \alpha w(A, B)$ si $u \in A$. En combinant les deux équations, on obtient :

$$w(A, B) + w(\{u\}, A) - w(\{u\}, B) \leq \alpha w(A, B)$$

□

Nous considérons un réel arbitraire $\epsilon > 0$. Soit n le nombre de sommets de G. Nous allons supposer par la suite que $\alpha = 1 + \frac{2\epsilon}{n}$.

Soit (A, B) la solution obtenue par l'algorithme.

Question 2.5 Montrer que

1. $\sum_{u \in A} w(\{u\}, A) \leq w(A, B) + \frac{|A|2\epsilon}{n} w(A, B)$
2. $\sum_{u \in B} w(\{u\}, B) \leq w(A, B) + \frac{|B|2\epsilon}{n} w(A, B)$
3. $W(A) + W(B) \leq (1 + \epsilon)w(A, B)$

Correction

En appliquant la question précédente ($w(\{u\}, A) \leq w(\{u\}, B) + (\alpha - 1)w(A, B)$), on obtient

$$w(\{u\}, A) \leq w(\{u\}, B) + \left(\frac{2\epsilon}{n}\right) w(A, B)$$

En sommant sur tous les sommets de A (resp. de B), on obtient la première (resp. deuxième) équation.

En sommant les deux inégalités précédentes (1) et (2), on obtient

$$\sum_{u \in A} w(\{u\}, A) + \sum_{u \in B} w(\{u\}, B) \leq 2w(A, B) + 2\epsilon \frac{|A| + |B|}{n} w(A, B) \qquad (2)$$

Remarquons que $\sum_{u \in A} w(\{u\}, A) = 2W(A)$ et $\sum_{u \in B} w(\{u\}, B) = 2W(B)$. Nous pouvons réécrire l'équation (**??**)

$$2W(A) + 2W(B) \leq 2w(A, B) + 2\epsilon w(A, B)$$

Donc, nous obtenons

$$W(A) + W(B) \leq (1 + \epsilon)w(A, B)$$

□

Question 2.6 Montrer que $W(E) \leq (2 + \epsilon)w(A, B)$.

Correction

Rappellons que $W(E) = W(A) + W(B) + w(A, B)$.

D'après la question précédente,

$$W(A) + W(B) + w(A, B) \leq (1 + \epsilon)w(A, B) + w(A, B).$$

Nous pouvons déduire que $W(E) \leq (2 + \epsilon)w(A, B)$.

□

Question 2.7 Montrer que $w(A^*, B^*) \leq (2 + \epsilon)w(A, B)$. Qu'en déduisez-vous ?

Correction

Il suffit de constater que $W(E) \geq w(A^*, B^*)$. Comme $(2+\epsilon)w(A, B) \geq W(E)$, nous en déduisons

$$(2 + \epsilon)w(A, B) \geq W(E) \geq w(A^*, B^*)$$

□

Soit (A_0, B_0) la coupe initiale utilisée par l'algorithme. Nous allons considérer que $A_0 = \{v^{max}\}$ et $B_0 = V \backslash A_0$ tel que le sommet v^{max} respecte la condition suivante

$$\forall v \in V, w(\{v^{max}\}, V \backslash \{v^{max}\}) \geq w(\{v\}, V \backslash \{v\}) \tag{3}$$

Question 2.8 Montrer que $w(A_0, B_0) \geq \frac{2}{n}W(E)$.

Correction

Notons $w(A_0, B_0) = w(\{v^{max}\}, V \backslash \{v^{max}\})$.
Rappellons que $\sum_{v \in V} w(\{v\}, V \backslash \{v\}) = 2W(E)$.
Par définition du sommet v^{max}, nous obtenons $nw(\{v^{max}\}, V \backslash \{v^{max}\}) \geq 2W(E)$.
Donc

$$nw(A_0, B_0) \geq 2W(E)$$

□

Soit (A_t, B_t) la coupe de l'algorithme après l'itération t de la boucle TANT QUE.

Question 2.9 Montrer que $w(A_{t+1}, B_{t+1}) \geq (1 + \frac{2\epsilon}{n})w(A_t, B_t)$

Correction

Si l'algorithme ne se termine pas après l'itération t de la boucle TANT QUE, alors il existe un sommet v_1 non *α-content*.

Supposons que $v_1 \in B_t$ (sans perte de généralité) et qu'il est sélectionné pour l'itération $t + 1$. De ces hypothèses, nous pouvons déduire que $\alpha w(A_t, B_t) < w(A_t \cup \{v_1\}, B_t \backslash \{v_1\})$

Comme $w(A_t \cup \{v_1\}, B_t \backslash \{v_1\}) = w(A_{t+1}, B_{t+1})$, on a

1. $\alpha w(A_t, B_t) < w(A_{t+1}, B_{t+1})$
2. $(1 + 2\frac{\epsilon}{n})w(A_t, B_t) < w(A_{t+1}, B_{t+1})$

□

Question 2.10 Donner une borne inférieure sur le nombre d'itérations k de l'algorithme pour que le poids de la coupe double $(w(A_{t+k}, B_{t+k}) \geq 2w(A_t, B_t))$. (*Indication : $(1 + 1/x)^x \geq 2$ pour $x \geq 1$*)

Comme on a $w(A_{t+1}, B_{t+1}) \geq (1 + \frac{2\epsilon}{n}) w(A_t, B_t)$, nous pouvons généraliser :

$$\begin{array}{rcl} w(A_{t+k}, B_{t+k}) & \geq & (1+\frac{2\epsilon}{n}) w(A_{t+k-1}, B_{t+k-1}) \\ & \geq & (1+\frac{2\epsilon}{n})^2 w(A_{t+k-2}, B_{t+k-2}) \\ & \geq & (1+\frac{2\epsilon}{n})^k w(A_t, B_t) \end{array}$$

Pour $x \geq 1$, nous avons $(1 + 1/x)^x \geq 2$. Ceci implique que $(1 + \frac{2\epsilon}{n})^{n/2\epsilon} \geq 2$.
Donc pour $k \geq \frac{n}{2\epsilon}$, nous avons $w(A_{t+k}, B_{t+k}) \geq 2w(A_t, B_t)$.

□

Question 2.11 Soit K le nombre de l'itérations nécessaire pour que l'algorithme termine. Montrer que $K \geq \frac{|V| log_2 W(E)}{\epsilon}$. Quelle est la complexité de l'algorithme ?

Correction

Notons l'entier ℓ tel que $2^\ell \geq W(E) \geq 2^{\ell-1}$ et tel que $\ell \geq \log_2 W(E) \geq \ell - 1$. Nous pouvons remarquer que $(1 + \frac{2\epsilon}{n})^{\ell \frac{n}{2\epsilon}} \geq 2^\ell$ d'après les questions précédentes.

Nous avons les conditions sur K tel que $K \geq \ell \frac{n}{\epsilon}$. Ceci implique que $(1 + \frac{2\epsilon}{n})^K \geq 2^\ell$, que $(1 + \frac{2\epsilon}{n})^K \geq W(E)$, et que $(1 + \frac{2\epsilon}{n})^K w(A_0, B_0) \geq W(E)$.

Donc, il y a au plus K itérations de la boucle.

□

Exercice 3 Ensemble dominant

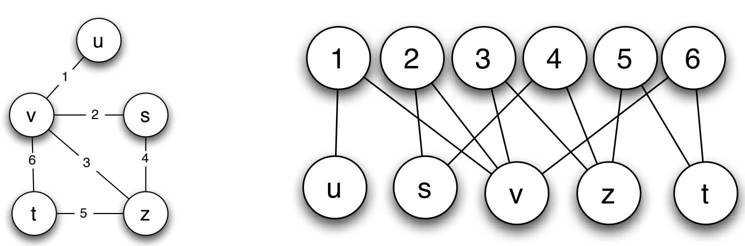
Donnons la définition des ensembles dominants : un *ensemble dominant* C du graphe G est un sous-ensemble de sommets tel que tout sommet est soit dans C soit voisin d'un sommet de C.

Soit un graphe $G = (V, E)$ ne possédant aucun sommet isolé. Nous allons construire un graphe $G' = (V', E')$ à partir de G tel que

— $V' = V \cup E$;

— $E' = E \cup \{(v, e) | v \in V, e \in E, v$ est extremité de l'arête e dans $G\}$

La figure 2 donne une illustration de cette construction.



Graphe G Graphe G'
(les arêtes de G' dans E ne sont pas dessinées)

FIGURE 2 – Graphes G' et G

Question 3.1 Montrer que si S est une couverture de sommets du graphe G, alors S est un ensemble dominant de G'.

Correction

Soit S une couverture de sommets du graphe G. Il faut prouver que tous les sommets du graphe soit soit voisin de S ou soit dans S.

Toutes les arêtes de G ont au moins un de ses extrémités dans S. Par conséquence, tous les sommets de G sont soit dans S ou soit voisins de S.

Comme toutes les arêtes de G ont au moins un de ses extrémités dans S, tous les sommets correspondant à une arête de G sont des voisins de S.

Donc S est un ensemble dominant de G'. $\square$

Question 3.2 Montrer que si S' est un ensemble dominant de G', alors il existe un ensemble S de même cardinalité de S' qui est une couverture de sommets du graphe G,

Correction

Si $S' \subseteq V$, alors tous les sommets du graphe G soit soit voisin de S ou soit dans S. Donc toutes les arêtes ont une de ces extrémités dans S.

Donc S' est une couverture de sommets du graphe G,

Sinon, il existe un sommet s qui n'appartient pas à V.

$\square$

Question 3.3 Exprimer le problème de minimisation de l'ensemble dominant sous forme de problème de décision.

Correction

Données : un graphe non-orienté G, et un entier k
Question : Existe-t-il un ensemble dominant S de G tel que $|S| \leq k$?

$\square$

Question 3.4 Montrer que ce problème est NP-complet.

Correction

On peut vérifier en temps polynomial si un ensemble de sommets est un ensemble dominant et si il est de cardinalité inférieure à k. $\square$

Correction

$\square$