ENTPE

ECOLE NATIONALE DES TRAVAUX PUBLICS DE L'ETAT

Groupe 5

Mini-Projet de MRO

Flots dans les graphes

EL HBARI Naïm – ENEZIAN Gustave

Table des matières

1) POSITION DU PROBLEME

L'objet du miniprojet est d'aborder des problèmes de flots dans les graphes. Deux angles d'approche seront utilisés, la programmation linéaire et l'algorithme de Ford-Fulkerson. La première partie du miniprojet est dédiée aux problèmes de flots sans contraintes de capacité. Elle permettra de faire le lien entre flots, chemins et programmation linéaire. La seconde partie du miniprojet est dédiée aux problèmes de flots avec contraintes de capacité et notamment à l'algorithme de Ford-Fulkerson. Celui-ci est particulièrement important pour l'interprétation des solutions et l'implication pratique de ces solutions. La troisième et dernière du miniprojet est dédiée à la résolution d'un problème de flots capacités par la programmation linéaire dans le cadre d'un problème de réseau de transports.

2) FLOTS NON CAPACITES : LIEN ENTRE CHEMIN, FLOT ET PROGRAMMATION LINEAIRE

a) Question 1

Par définition de la matrice A, chacune de ses colonnes possède seulement 2 termes non nuls $A_{k,kl}$ et $A_{l,kl}$ égal respectivement à +1 et -1.

Nous allons montrer par récurrence sur $n \in \mathbb{N}^*$ que A est totalement unimodulaire, n étant le nombre de lignes et de colonnes de la sous-matrice considérée, autrement dit on suppose que la sous-matrice A_n de A appartient à $\mathrm{M}_n(\mathbb{R})$.

- ✓ Amorce n=1 : dans ce cas le seul et unique terme de la sous-matrice vaut soit +1, soit -1, soit 0 donc son déterminant vaut respectivement +1, ou -1 ou 0.
  La propriété est donc vérifiée pour n=1.

- ✓ Passage n→ n+1 : Supposons que la propriété est vérifiée pour $n \in \mathbb{N}^*$. Montrons qu'elle rest vraie pour n+1.
  Soit A_{n+1} une sous matrice de A telle que $A_{n+1} \in \mathrm{M}_n(\mathbb{R})$.
  Nous allons chercher la colonne de A_{n+1} ayant le moins d'éléments non nuls, il y a cas :
  1) A_{n+1} possède une colonne avec uniquement des 0 :
     Dans ce cas det(A_{n+1})=0 donc la propriété est vérifiée au rang n+1.

  2) A_{n+1} possède une colonne avec uniquement un élément aij non nul (donc cet élément aij est égal à +1 ou -1) :
     Dans ce cas, si on calcule le déterminant de A_{n+1} en développant par rapport à cette colonne, on trouve

$\det(A_{n+1}) = (-1)^{i+j} aij.\det(A_n^{\ ij})$ or $A_n^{\ ij} \in \mathrm{M}_n(\mathbb{R})$ donc $\det(A_n^{\ ij}) = \begin{cases} 0 \\ +1 \\ -1 \end{cases}$ d'après l'hypothése de récurrence

⇨ $\det(A_{n+1}) = \begin{cases} 0 \\ +aij \\ -aij \end{cases} = \begin{cases} 0 \\ +1 \\ -1 \end{cases}$ donc la propriété est vérifiée au rang n+1

3) A_{n+1} possède deux éléments non nuls (donc +1 et -1) dans chacune de ses colonnes :

Dans ce cas, je somme chacune des lignes dans la dernière colonne (selon la méthode du pivot de Gauss) :

$$\begin{pmatrix} L_1 = L_1 \\ : \\ L_i = L_i \\ : \\ L_{n+1} = L_{n+1} + \sum_{i=1}^{n} L_i \end{pmatrix}$$

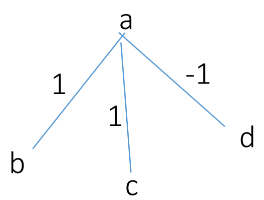
Dans la ligne L_{n+1}, pour chacune de ses colonne, la somme est égale à une somme de 0 +1 -1 = 0 (pour la j-eme colonne de L_{n+1} : l'élément $L_{n+1,j} = 0$ car tous les éléments sont nuls sauf deux éléments égaux +1 et -1 ce qui fait que la somme totale est nulle)

Donc $\det\left(A_{n+1}\right) = \det \begin{pmatrix} L_1 = L_1 \\ : \\ L_i = L_i \\ : \\ L_{n+1} = L_{n+1} + \sum_{i=1}^{n} L_i \end{pmatrix} = \det \begin{pmatrix} L_1 = L_1 \\ : \\ L_i = L_i \\ : \\ 0...0...0 \end{pmatrix} = 0$ car tous les éléments de la dernière ligne sont nuls, donc la propriété est vérifiée au rang n+1.

Conclusion : A est totalement unimodulaire

- ✓ Dans une base, il n'y a aucun cycle (sinon les colonnes seraient liées entre elles) donc toutes les colonnes de A sont linéairement indépendantes. A est donc un arbre par définition.

Voici un exemple de vecteur de base de A (comme nous l'avons vu avec Mr Lebecque pendant la séance de préparation au projet) :



Ce qui donne comme sous-matrice de A :

$$\begin{array}{cc} \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{pmatrix} 1 & 1 & -1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ & \begin{matrix} ab & ac & ad \end{matrix} \end{array}$$

Dans notre cas, les 3 colonnes sont indépendantes donc nous aurions pu prendre une sous-base B' de B faite d'arcs uniquement (B'=((ab),(ac),(ad)))

Dans une base B, A peut s'écrie comme une matrice A_B inversible de taille égale au nombre de vecteurs de la base (donc au nombre de nœuds)

L'écriture du système (3) dans une base peut s'écrire :

$$\begin{cases} \varphi_B = A_B^{-1}.b_B \\ \varphi_B \geq 0 \end{cases}$$

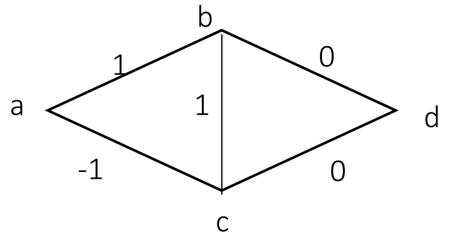
b) Question 2

- ✓ Si $Supp(\varphi)$ est un arbre et si $A\varphi = 0$, ceci signifie que la valeur de flux entrant et sortant de l'arbre est nul, donc par conservation du flot cela signifie que $\varphi = 0$ partout.
  En effet $b = 0 \Rightarrow b_{source} = 0 \quad et \quad b_{puit} = 0$ donc V=0 et par suit $\varphi_{ij} = 0$

D'après les résultats de la question précédente, nous retrouvons bien ce résultat car si A est un arbre, il existe une base B telle que $A_B.\varphi_B = 0$ avec A_B inversible donc $\varphi_B = 0$ d'où $\varphi = 0$
.

- ✓ Cherchons maintenant à déterminer un flot $\varphi \neq 0$ sur un cycle tel que $A\varphi = 0$:

Nous proposons l'exemple suivant :



Dans ce cas, nous avons bien un support non nul ($\varphi_{ab} = 1$ et $\varphi_{ac} = -1$) or le flux à la source a est égal à -1+1=0 ($b_{source} = 0$) et le flux au puit d=0 ($b_{puit} = 0$) donc V=0 et b=0.

Nous avons bien $A\varphi = 0$ et $\varphi \neq 0$. Voici la matrice d'incidence (pour le support non nul) :

$$\begin{array}{c} a \\ b \\ c \\ d \\ \end{array} \begin{array}{c} \begin{pmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{pmatrix} \\ \begin{matrix} ab & ac & bc \end{matrix} \end{array} \Rightarrow C_2 = -C_1 - C_3$$

Donc les colonnes ne sont pas linéairement indépendantes, nous retrouvons bien le résultat de la première question (lorsque A n'est pas inversible $A\varphi = 0$ n'implique pas $\varphi = 0$)

c) Question 3

Les bases du système (3) sont donc les ensembles $\{sommet + arcs\}$ tels que : les sommets $x \in G$ et les arcs (x,y) tels que $y \in Succ(x)$ et en ne prenant aucun cycle en compte. L'ensemble des arcs d'un vecteur de cette base forme une sous-base B' de ce vecteur (donc $B = \bigcup B'$)

Autrement dit, à partir d'un graphe connexe quelconque, on peut obtenir un arbre couvrant ou arbre maximal d'après le cours (partie sur l'algorithme de Kruskal) et à partir de cet arbre fait de n sommets, nous aurons n-1 arcs qui représenteront d'après les questions précédentes une base du système (3).

Le rang de A est égal au nombre de vecteurs colonnes linéairement indépendantes, donc par définition rg(A)=dim(base B)=n-1.

Une base réalisable est une base pour laquelle le flot entrant reste constant (égal à V) sur toute une branche de l'arbre jusqu'au puit (le flot étant nul ailleurs).

Autrement, si le flot se sépare pour se rejoindre à un nœud, on peut dire qu'il y a création d'un cycle donc nous ne serions plus en présence d'un arbre.

Dans ce type de base, on aura $b = \begin{pmatrix} V \\ : \\ : \\ -V \end{pmatrix}$

d) Question 4

D'après le théorème du nombre cyclomatique, pour un graphe contenant n sommets, m arcs, ayant p composantes connexes et c cycles indépendants. Les 4 nombres sont reliés par la relation :

p = n – m + c

donc le nombre de cycles indépendants c est égal p-n+m.

Dans un graphe, lorsque l'on augmente le nombre de cycles, on diminue la connexité p (cf démonstration du théorème du nombre cyclomatique) et donc le nombre maximal de cycles indépendants d'un graphe connexe est 1-n+m (p=1).

Ceci explique pourquoi la borne de l'indice p (parcourant les cycles ici, à ne pas confondre avec la connexité) est égale à m-n+1.

A chaque cycle, le flot Φ_p de support Γ_p satisfait à $A\Phi_p = 0$ (les composantes sont égales à 0, 1 et -1). Comme nous l'avons vu dans la question 2, ceci entraine que la valeur du flot entrant et sortant de chaque cycle est nul.

Si $A\varphi = 0$ alors $Supp(\varphi)$ n'est pas un arbre d'après 2), donc il s'agit forcement d'un graphe contenant au moins un cycle.

Par définition de Γ_p, qui constitue une base cycles indépendants, le support de φ est constitué d'éléments Φ_p de la base Γ_p parcours α_p fois chacun (on peut faire plusieurs fois le tour d'un cycle) et donc :

$$\varphi = \sum_{p=1}^{m-n+1} \alpha_p.\Phi_p$$

e) Question 5

Soit B une base réalisable de (3). On met les coûts réduits sous la forme $\gamma = uA - w$.

Nous pensons que que cette technique permettra de faire le lien avec l'algorithme du simplexe vu en cours et de pouvoir l'utiliser pour ce type de problème. Le fait de mettre un signe – devant w permettra le passage du Min au Max.

$$\begin{array}{l} Min \ \sum_{(k,l)\in U} w_{kl}\,\varphi_{kl} \\ \left| \begin{array}{l} A\varphi = b \\ \varphi \geq 0 \end{array} \right. \end{array} \rightarrow \begin{array}{l} \text{Max } c.x \\ \left| \begin{array}{l} A.x = b \\ x \geq 0 \end{array} \right. \end{array}$$

Formule du Problème — Formule du cours (chapitre du simplexe)

Dans la base B réalisable, A est inversible d'après ce qui précède donc $u = (\gamma + w)A^{-1}$

B est optimale si cette base permet de parcourir une branche de l'arbre depuis la source jusqu'au puit, donc B est optimale si le flot φ n'est jamais égal à 0 sur les éléments de la base.

U pourra se calculer par une méthode de pivotage vue en cours.

U correspond à des poids maximaux ; c'est la raison pour laquelle l'algorithme du simplexe peut s'appliquer ici.

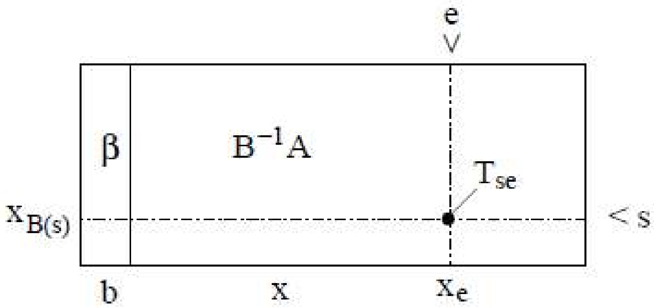
f) Question 6

La colonne (kl) de A dans la base B est une colonne avec uniquement des 0, excepté un 1 à la k-éme ligne et un -1 à la l-éme ligne.

On choisit la ligne sortante de la même manière que pour le simplexe :

$$\left(\begin{array}{lcll} B'(s) & = & e & \\ B'(i) & = & B(i) & \text{si } i \neq s \end{array} \right. \tag{13}$$

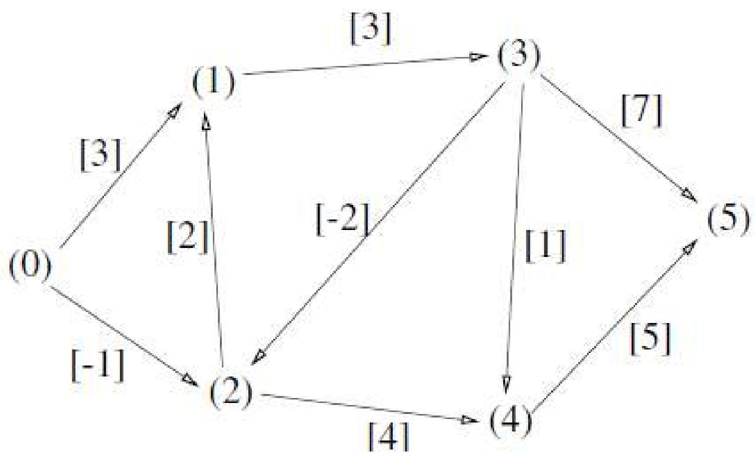
ce que nous illustrerons comme suit:



La colonne e s'appelle *colonne entrante*, et la ligne s s'appelle *ligne sortante*.

C'est donc la k-éme ligne dans notre cas.

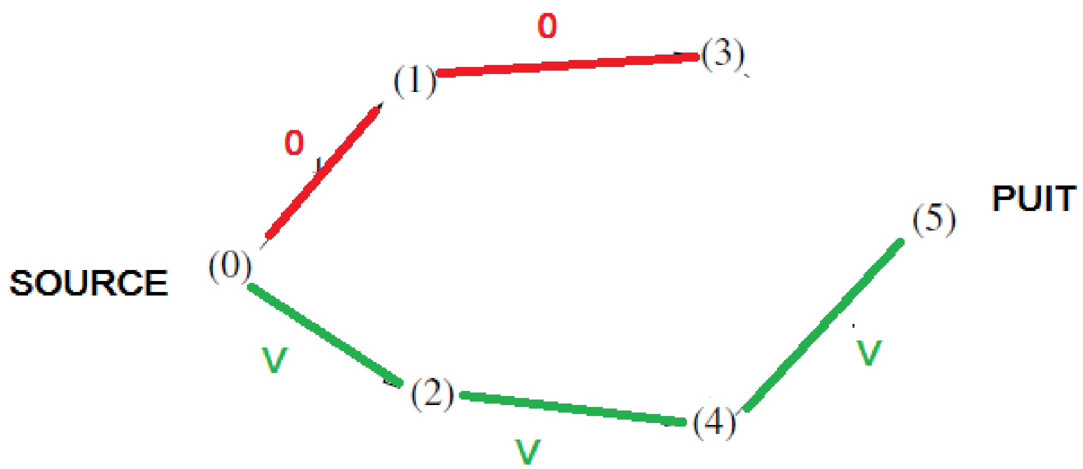
g) Question 7



En partant du graphe précédent, nous ne gardons que les arcs présents dans la base B.

La base de départ formée des arcs (0,1), (0,2), (1,3), (2,4), (4,5) est bien réalisable car elle permet de relier la source au puit par une branche (0-2-4-5).

Il n'y a qu'une solution réalisable dans la base B donc il n'est pas utile de calculer le poids minimal (=5+4-1=8)



La branche figurant en rouge ne débouche pas sur le puit c'est la raison pour laquelle la valeur du flot est nulle.

Automatiquement, la valeur V est attribuée au flot parcourant l'autre branche de la base (en vert) allant de la source (0) au puit (5).

h) Question 8

Etant donné qu'il y a existence d'un flot V non nul uniquement si le graphe est un arbre, la solution optimale de (3) consiste donc uniquement à déterminer un des chemins (branche) amenant de la source au puit. Le fait de ne pas avoir de cycles facilite fortement la tâche.

Lorsque plusieurs chemins sont possibles, il suffit de choisir celui de poids minimal.

La condition nécessaire et suffisante d'existence d'un tel chemin est que le flot $\varphi \neq 0$ et qu'il y ait un chemin reliant la source au puit.

B sera donc optimale si son rang est égal à n-1 (donc inversible).

3) INTRODUCTION AUX PROBLEMES DE FLOTS CAPACITES : ALGORITHME DE FORD-FULKERSON

a) Théorème de la coupe optimale

Soit une coupe $(Y,\bar{Y})$ de l'ensemble des sommets X.
Notons $f(Y,\bar{Y})$ le flot de la coupe passant de Y à $\bar{Y}$.

Montrons que $V(f) = f(Y,\bar{Y}) - f(\bar{Y},Y)$
On raisonne par récurrence sur le cardinal de Y.

- Initialisation :

Card(Y)=1
Alors $Y = \{s\}$
Ainsi $f(Y,\bar{Y}) = \sum_{x\in\Gamma^+(s)} f(s,x) = V(f)$ (car s n'a pas de prédécesseurs dans Y) et $f(\bar{Y},Y) = 0$

- Hérédité :

Soit $Y' = Y + \{x\}$

$$f(Y',\bar{Y'}) - f(\bar{Y'},Y')$$

$$= f(Y \cup \{x\}, \bar{Y}\backslash\{x\}) - f(\bar{Y}\backslash\{x\}, Y \cup \{x\})$$

$$= f(Y,\bar{Y}) - f(\bar{Y},Y) + \underbrace{\underbrace{f(x,\bar{Y}) + f(x,Y)}_{f(x,U)} \underbrace{- f(Y,x) - f(\bar{Y},x)}_{f(U,x)}}_{=0}$$

D'où l'hérédité.

- Conclusion : $V(f) = f(Y,\bar{Y}) - f(\bar{Y},Y)$

Or, $\begin{cases} f(Y,\bar{Y}) \leq c(Y,\bar{Y}) \; car \; \forall y \in Y \; \forall z \in \bar{Y}, f(y,z) \leq c(y,z) \\ f(Y,\bar{Y}) \geq 0 \end{cases}$

D'où $f(Y,\bar{Y}) - f(\bar{Y},Y) \leq c(Y,\bar{Y})$
Soit $V(f) \leq c(Y,\bar{Y})$
Désormais supposons que $V(f) = c(Y,\bar{Y})$ ie $\begin{cases} f(Y,\bar{Y}) = c(Y,\bar{Y}) \\ f(Y,\bar{Y}) = 0 \end{cases}$

Comme le flot a la valeur de la capacité de la coupe, il est maximal car par définition : tous les flots sont inférieurs à la capacité de n'importe quelle coupe.

Il s'agit donc par ailleurs de la coupe de capacité minimale.

b) Chaîne augmentant le flot

Etudions les 3 cas suivants :

Premier cas : $u \notin c$ alors $f'(u) = f(u)$ donc f' est un flot admissible.

Deuxième cas : $u = u_i = (x_{i-1}, x_i)$

Ainsi $f'(u) = f(u) + \kappa(c) = f(u) + Min(c(u_i) - f(u_i))$:

- Si $u = u_i$ alors $f'(u) = c(u)$
- Sinon $f'(u) \leq c(u)$ car $f(u)$ est un flot admissible.

D'où $0 \leq f'(u) \leq c(u)$

Troisième cas : $u = u_i = (x_i, x_{i-1})$

Alors $f'(u) = f(u) - Min(f(u_i))$:

- Si $Min(f((u_i))) = f(u)$, alors $f'(u) = 0 \leq c(u)$
- Sinon $f'(u) > 0$ et $f'(u) \leq c(u)$ car $f(u)$ est un flot admissible.

c) L'algorithme de Ford Fulkerson : marquage

- Si le marquage s'arrête

Soit une coupe $(Y, \bar{Y})$ comme le décrit l'énoncé. Sa capacité vaut : $c(Y, \bar{Y}) = \sum_{x \in Y, y \in \bar{Y}} c(x, y)$

On va montrer que $c(Y, \bar{Y}) = f(Y, \bar{Y})$

Soit $\begin{cases} x \in Y \\ y \in \bar{Y} \end{cases}$ tels que l'arc (x,y) existe.

Si lors de l'itération, y n'est pas marqué, cela signifie que :

-soit $y \in \Gamma^+(x)$ et donc l'arc (x,y) est saturé ($f(x, y) = c(x, y)$).

-soit $y \in \Gamma^-(x)$ et donc $f(y, x) = 0$

Comme $y \in \bar{Y}$ quand l'algorithme se termine, on obtient donc : $f(Y, \bar{Y}) = c(Y, \bar{Y})$

$f(Y, \bar{Y}) = 0$

Et ainsi $(Y, \bar{Y})$ est une coupe optimale d'après la partie 1. f est alors maximal.

- Si t est marqué

Alors le flot f n'atteint jamais c.

On trouve alors une chaîne qui augmente le flot afin qu'avant t, on puisse trouver une coupe notée $(Y, \bar{Y})$comme vu précédemment.

d) Exercice de maniement

Etape 1 : $Y = \{a\}$ et $\bar{Y} = \{b, c, d, e, f, g\}$

Etape 2 : $Y = \{a, b = \mathcal{M}(a)\}$ et $\bar{Y} = \{c, d, e, f, g\}$
Etape 3 : $Y = \{a, b = \mathcal{M}(a), d = \mathcal{M}(b)\}$ et $\bar{Y} = \{c, e, f, g\}$
Etape 4 : $Y = \{a, b = \mathcal{M}(a), d = \mathcal{M}(b), c = \mathcal{M}(d)\}$ et $\bar{Y} = \{e, f, g\}$
Fin : car e, f et g sont inaccessibles.
La coupe optimale est donc : $(\{a, b, c, d\}, \{e, f, g\})$
Et on a donc comme capacité de la coupe :

$$c(Y, \bar{Y}) = \sum_{y \in Y, z \in \bar{Y}} c(y, z) = c(b, f) + c(c, e) + c(d, e) + c(d, f)$$

Ainsi $c(Y, \bar{Y}) = 30$

Puis on change : c(d,f)=12

Etape 1 : $Y = \{a\}$ et $\bar{Y} = \{b, c, d, e, f, g\}$
Etape 2 : $Y = \{a, b = \mathcal{M}(a)\}$ et $\bar{Y} = \{c, d, e, f, g\}$
Etape 3 : $Y = \{a, b = \mathcal{M}(a), d = \mathcal{M}(b)\}$ et $\bar{Y} = \{c, e, f, g\}$
Etape 4 : $Y = \{a, b = \mathcal{M}(a), d = \mathcal{M}(b), c = \mathcal{M}(d), f = \mathcal{M}(d)\}$ et $\bar{Y} = \{e, g\}$
Etape 5 : $Y = \{a, b = \mathcal{M}(a), d = \mathcal{M}(b), c = \mathcal{M}(d), f = \mathcal{M}(d), g = \mathcal{M}(f)\}$ et $\bar{Y} = \{e\}$
Etape 6 : $Y = \{a, b, c, d, f, g, e = \mathcal{M}(g)\}$ et $\bar{Y} = \{\emptyset\}$
Le flot n'est plus optimal (dès l'étape 5 car g est atteint.
Une chaîne augmentant le flot est :

$$c = ((a, b), (b, d), (d, f), (f, g))$$