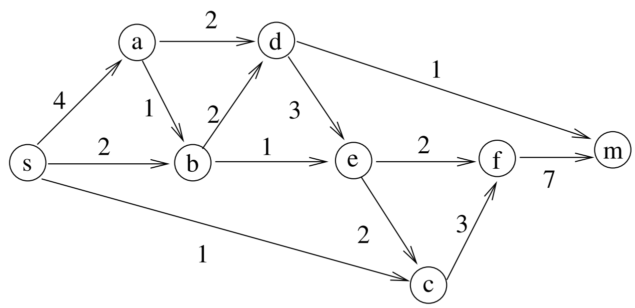
Algorithmique II
Année 2006-2007

Florent Bouchez
Jean-François Pineau

TD n^{o}8
Déluges et Flots

1 Retour au lit ?



Le cours d'une rivière, de la source à la mer, ne suit pas forcement un seul lit. Il se peut qu'elle suive plusieurs branches avant d'arriver à la mer, chaque branche ayant un débit propre. On suppose ici que la chaleur n'est pas suffisante pour que l'évaporation soit visible, de même que l'infiltration dans le sol ! ! Et, étant dans un milieu isolé, la rivière n'a pas d'autre alimentation en eau, la région étant faite de telle sorte qu'il ne pleuve qu'au niveau de la source.
Bref, rien ne se perd, rien ne se créé, tout part de la source pour aller à la mer.

On peut modéliser cela à l'aide d'un graphe orienté $G = (X, U)$, où s est la source et m la mer, et d'une fonction de valuation c indiquant la capacité (ou le débit) maximale des différentes branches de la rivière avant débordement.

Actuellement, la rivière s'écoule entre la source et la mer sans débordement, et peut être représentée par un flot φ sur G.

Un ancien L3IF, engagé par les gens des Eaux et Forêts, doit déterminer la quantité d'eau maximale pouvant s'écouler de la source à la mer afin de prévenir des risques d'inondation.

Exercice 1 Afin de répondre à son problème, notre ancien L3If prend les notes suivantes :

- Que peut-on dire de la fonction suivante : $c_\varphi(u) = c(u) - \varphi(u)$?
- Que peut-on dire du réseau suivant : $R_\varphi = (X, U_\varphi, c_\varphi)$ où $U_\varphi = \{(x, y) \in X \times X | c_\varphi(x, y) > 0\}$?
- Que représente un chemin μ de s à m dans ce réseau ?
- Que représente $c_\varphi(\mu) = \min\{c_\varphi(u) | u \in \mu\}$?
- Qu'est ce que cela implique sur le flot φ s'il n'existe pas de tel chemin ?

Exercice 2

Si on vous dit qu'après toutes ces questions, l'ancien L3IF est capable de résoudre ce problème, en êtes-vous capable aussi ?

Donc trouvez un algorithme pour résoudre le problème ! !

Voyez-vous un problème à votre algorithme ?

Voici l'algorithme de notre ancien L3IF :

Initialiser le flot φ à 0;
tant que *il existe un chemin améliorant de plus court chemin* μ *dans le graphe des écarts* **faire**
  améliorer φ le long de μ;
fin

Exercice 3

Soit φ' le flot obtenu juste après avoir augmenté le flot φ grâce au plus court chemin améliorant μ. En notant $d_\varphi(s,x)$ la distance (en nombre d'arcs) de s à x dans R_φ, montrer qu'après chaque passe dans la boucle, on a

$$\forall x \in X,\ d_\varphi(s,x) \leq d_{\varphi'}(s,x)$$

Exercice 4

En déduire que la boucle de l'algorithme est exécutée de l'ordre de $O(n'n)$ fois, où n est le nombre de sommets du réseau et n' le nombre d'arcs.

Exercice 5

En déduire que l'algorithme calcule un flot maximum en temps $O(nn'^2)$.

2 Une petite coupe avant le bain ?

Une *coupe* dans $R = (X, U, c)$ est une partition de X en $(Y, \bar{Y})$ avec $s \in Y$ et $m \in \bar{Y}$. La *capacité* de la coupe est $c(Y, \bar{Y}) = \sum_{y \in Y} \sum_{z \in \bar{Y}} c(y, z)$.

Exercice 6

Montrer que pour toute coupe $(Y, \bar{Y})$ d'un réseau R et tout flot φ sur R, on a $|\varphi| = \varphi(Y, \bar{Y})$. En déduire que $|\varphi| \leq c(Y, \bar{Y})$.

Exercice 7

Soit $C = \max\{c(x,y) | (x,y) \in U\}$. Montrer qu'une coupe minimum a une capacité au plus égale à $C|U|$.

Exercice 8

Soit φ un flot, montrer que les trois propositions suivantes sont équivalentes :
(i) φ est un flot maximum (càd de valeur maximum) ;
(ii) il n'existe pas de chemin améliorant ;
(iii) il existe une coupe $(Y, \bar{Y})$ telle que $|\varphi| = c(Y, \bar{Y})$.

Exercice 9

Donner un algorithme qui, étant donné un flot maximum φ calcule une coupe de capacité minimale.

3 Concurrent

Un autre ancien L3IF, concurrent sur cette affaire, propose l'algorithme suivant :

$C \leftarrow \max\{c(x,y) | (x,y) \in U\}$;
Initialiser le flot φ à 0;
$k \leftarrow 2^{\lfloor \log C \rfloor}$;
tant que $k \geq 1$ **faire**
  tant que *il existe un chemin améliorant* μ *de capacité au moins* k **faire**
    augmenter le flot le long de μ
  fin
  $k \leftarrow k/2$;
fin

Exercice 10 Pour un entier k donné, montrer que l'on peut calculer en $O(n + n')$ un chemin améliorant de capacité au moins k si un tel chemin existe.

Exercice 11 Prouvez que l'algorithme calcule effectivement un flot maximum.

Exercice 12

1. Montrer que la capacité résiduelle d'une coupe minimum est au plus $2kn'$ à chaque étape de la boucle *tant que* externe.
2. Montrer que la boucle *tant que* interne est exécutée au plus $O(n')$ fois pour chaque valeur de k.
3. Déterminer la complexité de l'algorithme.

4 Aménagement du territoire

Tout va bien, notre L3IF a enfin calculé le flot de capacité maximum... oui mais voila :

Exercice 13 Aux abords d'une grande ville, des travaux d'élargissement de l'autoroute ont comblé une partie de la rivière. Cette portion de la rivière (x, y) a donc maintenant une capacité décrémentée de 1.

Donner le principe d'un algorithme en temps linéaire pour mettre à jour le flot maximum. Justifier la complexité.

Exercice 14 Pour réparer les dégâts, on utilise des engins de chantier pouvant creuser le lit de la rivière. Seulement, à la fin des travaux, on remarque que la capacité de la portion a été augmentée de 1 par rapport à la valeur initiale.

Donner le principe d'un algorithme en temps linéaire pour mettre à jour le flot maximum. Justifier la complexité.