Algorithmique II
Année 2006-2007

Florent Bouchez
Jean-François Pineau

TD n°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 ?

Solution:
- *c'est la capacité résiduelle d'un lien*
- *c'est le graphe des écarts*
- *un chemin représent un chemin améliorant pour le flot*
- *c'est la capacité du chemin améliorant*
- *le flot est maximal, car il ne peut plus être amélioré*

Exercice 2

Après toutes ces questions, l'ancien L3IF est capable de résoudre ce problème. Voici l'algorithme :

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

Voyez-vous un problème à votre algorithme ?

Solution: *Le temps d'éxécution peut être extrèmement long, selon les poids des chemins, si on choisit mal le chemin améliorant (voir le cours)*

Exercice 3 A présent, notre L3IF choisit de prendre comme chemin améliorant le plus court chemin dans le graphe des écarts.

Appliquer cet algorithme au graphe donné.

Solution: *Chemins améliorants :* $sadp, scfp, sbefp, sabefp, sdbecfp, sabdecfp$.

Exercice 4

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

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

Solution: *Supposons qu'il existe* $x \in X \backslash \{s,p\}$ / $d_{\varphi'}(s,x) < d_\varphi(s,x)$ *et* $d_{\varphi'}(s,x)$ *soit minimal. Considérons alors* μ' *le plus court chemin de* s *à* x *réalisant* $d_{\varphi'}(s,x)$*, et appelons* y *le prédécéseur de* x *sur* μ'*.*

On a $d_{\varphi'}(s,x) = d_{\varphi'}(s,y) + 1 \leq d_\varphi(s,y) + 1$.

- *Si* $\varphi(y,x) < c(y,x)$*, alors* $(y,x) \in U_\varphi$.
  On a alors $d_\varphi(s,x) \leq d_\varphi(s,y) + 1 \leq d_{\varphi'}(s,y) + 1 \leq d_{\varphi'}(s,x) + 1$.
- *Si* $\varphi(y,x) = c(y,x)$ *alors* $(y,x) \notin U_\varphi$*. Mais* $(y,x) \in U_{\varphi'}$*, donc* $(x,y) \in \mu$*. Or* μ *est le plus court chemin de* s *à* x*, donc* $d_\varphi(s,x) = d_\varphi(s,y) - 1 \leq d_{\varphi'}(s,y) - 1 \leq d_{\varphi'}(s,x) - 2$

Exercice 5

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.

Solution: *Un arc* (x,y) *de* G_φ *est critique sur un chemin améliorant* μ *si* $c_\varphi(x,y) = c_\varphi(\mu)$*. Evaluons le nombre de fois où un arc* (x,y) *est critique.*

Supposons (x,y) *critique sur* μ *pour* G_φ*. On a* $d_\varphi(s,y) = d_\varphi(s,x) + 1$*. Lorsque le flot est augmenté, l'arc* (x,y) *disparaît du graphe des écarts. L'arc* (x,y) *reapparaîtra lorsque l'arc* (y,x) *sera sur le chemin améliorant. Appelons* φ' *le flot tel que* (y,x) *est sur le chemin améliorant.*

$$d_{\varphi'}(s,x) = d_{\varphi'}(s,y) + 1 \geq d_\varphi(s,y) + 1 = d_\varphi(s,x) + 2$$

Si bien qu'entre les 2 fois où l'arc (x,y) *est critique, la distance de* s *à* x *est augmentée de 2.*

Au départ, $d(s,x) \geq 1$*, à la fin* $d(s,x) \leq n-2$*. Donc* (x,y) *est critique au plus* $O(n)$ *fois. Durant l'algo, il y a au plus* $O(nn')$ *arcs critiques. Comme tout chemin améliorant contient 1 arc critique, cela conclu la preuve.*

Exercice 6

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

Solution: *La boucle est éxécutée* $O(nn')$ *fois, et l'amélioration du flot le long du chemin améliorant dépent de la longeur du chemin, inférieure au nombre d'arêtes. La complexité totale est donc en* $O(nn'^2)$*.*

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 7

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 $|\varphi| \leq c(Y, \bar{Y})$.

Solution:

$$\begin{array}{rcll} \varphi(Y,\bar{Y}) & = & \varphi(Y,X) - \varphi(Y,Y) & = \textit{0 car chaque arc est vu dans les 2 sens} \\ & = & \varphi(Y,X) & \\ & = & \varphi(\{s\},X) + \varphi(Y \backslash \{s\}, X) & \forall\ y \in Y \backslash \{s\}\ \varphi(y,X) = 0 \Rightarrow \varphi(Y\backslash\{s\},X) = 0 \\ & = & \varphi(\{s\},X) & \\ & = & |\varphi| & \end{array}$$

$\forall (x, y) \in (Y, \bar{Y}), \varphi(x, y) \leq c(x, y) \Rightarrow \varphi(Y, \bar{Y}) \leq c(Y, \bar{Y})$

Exercice 8

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

Solution: *Pour toute arête $(x, y), x \in Y, y \in \bar{Y}$, $c(x, y) \leq C$ et il y a au plus $|U|$ arêtes entre Y et $\bar{Y}$.*

Exercice 9

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 $|\varphi| = c(Y, \bar{Y})$.

Solution:

($i \Rightarrow ii$) *Par l'absurde, s'il existe un chemin améliorant, le flot peut être amélioré, et donc il n'est pas maximum.*

($ii \Rightarrow iii$) *Posons $Y = \{x \in X \ / \ \exists\ \mu$ de s à x dans $G_\varphi\}$. Alors $s \in Y$ et $p \in \bar{Y}$ (puisque qu'il n'existe pas de chemin améliorant). De plus, $\forall (y, z) \in (Y, \bar{Y}), \varphi(y, z) = c(y, z)$, car sinon, $c_\varphi(y, z) > 0$, donc $(y, z)\ U_\varphi \Rightarrow z \in Y$, ce qui est une contradiction.*

($iii \Rightarrow i$) *D'après l'exercice précédent, $|\varphi| \leq c(Y, \bar{Y})$. Donc s'il existe une coupe réalisant l'égalité, c'est que le flot ne peut plus être augmenté ; il est maximum.*

Exercice 10

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

Solution:

Soit φ_{max};
Soit $G_{\varphi_{max}} = (X, U_{\varphi_{max}})$ le graphe des écarts;
Retirons toutes les arêtes saturées : $G'_{\varphi_{max}} \leftarrow (X, \{e \in U_{\varphi_{max}} \ / \ \varphi_{max}(e) < c(e)\})$;
Retourner la composante connexe de G' comprenant de s;

$C \leftarrow \max\{c(x,y) | (x,y) \in U\}$;
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

3 Concurrent

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

Exercice 11 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.

Solution: *Si le chemin a une capacité au moins k, c'est qu'il ne passe que par des arêtes de capacité supérieure.*

Initialiser le flot φ à 0;
Soit $G_\varphi = (X, U_\varphi)$ le graphe des écarts;
Retirons toutes les arêtes de capacité insuffisante : $G_{k,\varphi} \leftarrow (X, \{e \in U_\varphi \ / \ c(e) \geq k\})$;
Retourner le plus court chemin de s à m dans $G_{k,\varphi}$;

Exercice 12 Prouvez que l'algorithme calcule effectivement un flot maximum.

Solution: *La dernière occurrence de la boucle interne peut se réécrire :* tant qu'il existe un chemin améliorant de capacité au moins 1. *Donc tant qu'il existe encore de chemin améliorant, on les prends en compte. Au final, on obtient bien un flot maximum.*

Exercice 13

1. Montrer que la boucle *tant que* interne est exécutée au plus $O(n')$ fois pour chaque valeur de k.

   Solution: *Lorsqu'on visite une arête, elle a une capacité comprise entre k et $2k$, et une fois le flot amélioré le long de ce chemin, elle a une capacité inférieure à k. Chaque arête est donc visitée au plus une fois.*

2. Montrer que la capacité résiduelle d'une coupe minimum est au plus $2kn'$ à chaque étape de la boucle *tant que* externe.

   Solution: *Montrons le par récurrence sur le nombre d'étape de la boucle.*

A la première étape, la relation est vrai d'après l'exercice 8.
On se place à la i^{me} étape de la boucle tant que *externe.La boucle* tant que *interne peut se dérouler au plus n' fois, et chaque chemin améliorant trouvé est de capacité au plus $2k$ (puisque les arêtes de capacités supérieures ont été retiré lors du passage de la boucle externe précédente). La capacité résiduelle de la coupe minimum est donc au plus :*

$$\begin{aligned} c_i(Y,\bar{Y}) &\leq c_{i-1}(Y,\bar{Y}) - 2kn' \\ &\leq 2(2k)n' - 2kn' \\ &\leq 2kn' \end{aligned}$$

3. Déterminer la complexité de l'algorithme.

Solution:

Calcul de C	*max d'un tableau à m éléments*	$O(m)$
Initialisation de φ		$O(n+m)$
Boucle interne	*+ augmenter le flot*	$O(n+m)$
Boucle externe		$O(\log C)$
Complexité totale		$O(m(n+m)\log C)$

4 Aménagement du territoire

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

Exercice 14 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.

Solution: *Si l'arête n'était pas saturé avant les travaux, il n'y a pas de modification à faire. Sinon, on diffuse la contrainte physique sur le flot (le flot entrant est égale au flot sortant), ce qui décrémente le flot de 1. On cherche ensuite sur le nouveau graphe des écarts un chemin améliorant. Une seule étape suffit pour avoir alors un flot maximal.*

Exercice 15 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.

Solution: *On calcule le nouveau graphe des écarts, et on cherche un chemin améliorant. S'il en existe un, alors on améliore le flot le long de ce chemin. Comme le flot peut être au maximum amélioré de 1, une seule étape suffit.*