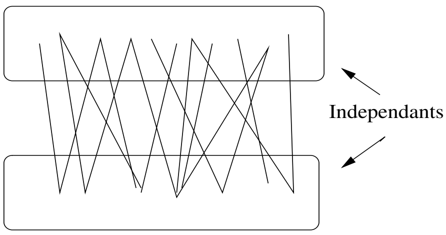
Flots dans les réseaux

Scribes : Rémy Bergasse Fabien Benureau

28 avril 2007

1 Étude des couplages dans les graphes bipartis

Définition 1.1 *Un graphe non orienté $G = (V, E)$ est biparti si on peut partitionner V en X et Y tq $E \subseteq xy, x \in X, y \in Y$*



Remarque : Ce sont les graphes 2-coloriables, ainsi que les graphes sans cylce de longueur impaire.

Définition 1.2 *Un couplage de G est un ensemble d'arètes $C \subseteq E$ tel que deux arètes de C n'ont jamais de sommet en commun.*

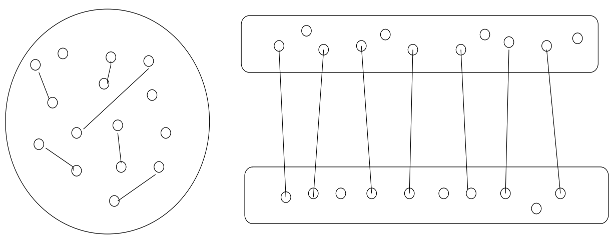


Fig. 1 – Couplage d'un graphe

Définition 1.3 *Soit C un couplage de $G = (V, E)$, et $x \in V$. On dit que x est* saturé *par C si x est l'extrémité d'une arète de C. Sinon, on dit que x est* insaturé.

Définition 1.4 *Un couplage C de G est* parfait *s'il sature tous les sommets de G.*

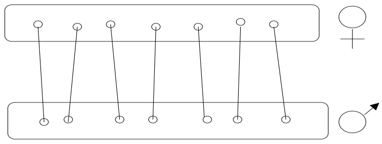


Fig. 2 – Couplage parfait

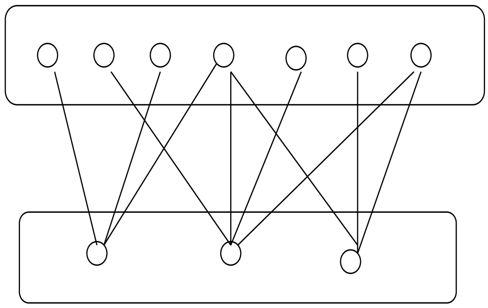


FIG. 3 – Exemple d'application

Problème classique : Trouver un couplage de cardinal maximum

Complexité : Dans le cas général, polynomial pour G quelconque. *Dans le cas biparti*, on peut se ramener au problème de flot maximum.

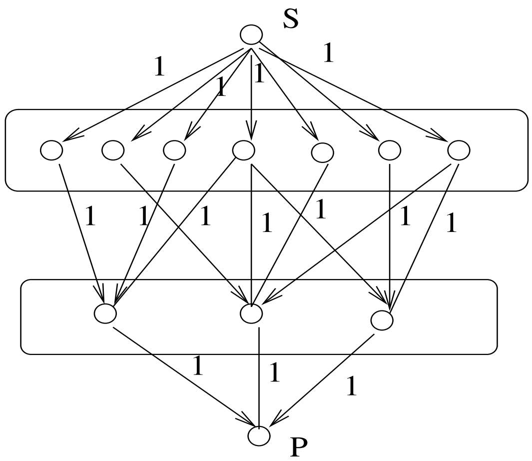


FIG. 4 – On ajoute une source et un puit, avec capacité de 1 partout.

Propriété 1.1 *Si les capacités du réseau sont dans $\mathbb{N}$, alors la valeur du flot maximum est dans $\mathbb{Z}$ et il existe un flot ϕ avec cette valeur tel que $\forall(x, y), \phi(x, y) \in \mathbb{Z}$*

Preuve :
- la valeur du flot maximum est dans $\mathbb{N}$
-

2 Algorithmes de flots

2.1 Algorithme de pré-flots (Goldberg)

2.1.1 Définition

Définition 2.1 *(pré-flot) $R = (V, E, c)$ réseau de source s et puits p. Un pré-flot est un application $\phi : E \to \mathbb{R}$ vérifiant :*
- *Pseudo-symétrie : $\forall x, y \phi(y, x) = -\phi(x, y)$*
- *Contrainte de capacité : $\forall (x, y) \in E, \phi(x, y) \leq c(x, y)$*
- *Relaxation de la loi de conservation : $\forall x \in X$*
  $\{\}, \phi(x, y) \leq c(x, y)$

Pour x *fixé*,

$$\sum_{(x,y)\in E} \phi(y,x) \geq \sum_{(x,y)\in E} \phi(x,y)$$

Flot en excès en x, noté $e(x)$, $e(x) = \phi(X,x)$ Un sommet x est dit **excédentaire** si $e(x) > 0$.

2.1.2 Opérations élémentaires

Définition 2.2 *Soit $R = (V,E,c)$ un réseau de source s et de puits p. Une* **fonction de hauteur** *h pour un préflot ϕ est un application $V \to \mathbb{N}$ vérifiant :*
- $h(s) = |V|$
- $h(p) = 0$
- $\forall (x,y) \in E_\phi, h(x) \leq h(y) + 1$

Opération pousser(x,y) . La poussée du flot à travers l'arc (x,y) applicable uniquement si
- $e(x) > 0$ (x excédentaire)
- $C_\phi(x,y) > 0$ ((x,y) pas saturé par ϕ)
- $h(x) = h(y) + 1$ (poussée du haut vers le bas)

Fonction `pousser(x,y)`.
- $d_\phi \leftarrow min(e(x), C_\phi(x,y))$;
- $\phi(x,y) \leftarrow \phi(x,y) + d_\phi$;
- $\phi(x,y) \leftarrow -\phi(x,y)$;
- $e(x) \leftarrow e(x) - d_\phi$
- $e(y) \leftarrow e(y) + d_\phi$

`pousser` est **saturante** si, après la poussée, l'arc (x,y) est saturé (ie, $c_\phi = 0$).

Opération élever(x) , applicable uniquement si :
- $e(x) > 0$
- $\forall (x,y) \in E_\phi, h(x) \leq h(y)$
- $x \neq p$

Fonction `pousser(x,y)`.
- $h(x) \leftarrow 1 + min\{h(y), (x,y) \in E_\phi\}$;

2.1.3 Algorithme générique

Initialisation
- pour tout $x \in V$, $h(x) \leftarrow 0$ et $e(x) \leftarrow 0$
- pour tout $(x,y) \in E$, $\phi(x,y) \leftarrow 0$ et $\phi(y,x) \leftarrow 0$
- $h(s) \leftarrow |V|$
- pour tout $x \in \mathbb{N}^+(s)$, $\phi(s,x) \leftarrow c(s,x)$ et $\phi(x,s) \leftarrow -\phi(s,x)$ et $e(x) \leftarrow \phi(s,x)$

Corps
- tant qu'il existe une opération de poussée ou d'élévation applicable, choisir une opération valide et l'effectuer.

Lemne 2.1 *Soit $R = (V,E,c)$ un réseau de source s, de puit p, de flot ϕ et de hauteur h. Si x est excédentaire, alors on peut appliquer au moins une opération élémentaire.*

Preuve $\forall (x,y) \in E_\phi, h(x) \leq h(y) + 1$.

1er cas : $\exists y, (x,y) \in E_\phi$ et $h(x) = h(y) + 1$. On peut appliquer à (x,y) une poussée.

2eme cas : $\forall y, (x,y) \in E_\phi, h(x) \leq h(y)$. On peut appliquer élever(x).

2.1.4 Correction de l'algorithme

Lemne 2.2 *Durant l'algorithme, les hauteurs ne décroissent jamais. Chaque fois qu'un sommet est elevé, sa hauteur augmente d'au moins 1.*

Lemne 2.3 *A chaque étape de l'algorithme, la fonction h reste une fonction de hauteur.*

Preuve On raisonne par récurrence sur le nombre d'étapes déjà effectuées.
- Après l'initialisation, c'est OK.
- Après une nouvelle étape,

1er cas : On a effectué une opération `élever(x)`. $x \neq p \neq s$. Si $(x, y) \in E_\phi$, $h(x) \leq h(y)+1$ après élévation. Si $(y, x) \in E_\phi$, avant élévation on avait $h(y) \leq h(x) + 1$. Vu que $h(x)$ augmente, on garde $h(y) \leq h(x) + 1$

2eme cas : On a effectué une opération `pousser(x,y)`. Avant l'opération, on avait forcément $h(x) = h(y) + 1$. Après l'opération, il faut vérifier que l'apparition d'arc dans E_ϕ ne change pas la propriété de hauteur. Le seul arc pouvant apparaître est (y, x). Dans ce cas, on sait que $h(y) = h(x) - 1 \leq h(x) + 1$.

Lemne 2.4 *Mêmes notations. Il n'existe pas de chemin de s à p dans R_ϕ.*

Preuve (par l'absurde). Supposons qu'il existe $\mu = (s = x_0, x_1, ..., x_k = p)$ un chemin élémentaire de s à p. Alors $\forall i, h(x_i) \leq h(x_{i+1})$. D'où $|V| = h(x_0) \leq h(x_k) + k < |V|$, impossible.

Théorème 2.1 *Si l'algorithme termine, il calcule un flot maximum.*

Preuve L'algorithme termine signifie qu'il n'y a plus d'opération applicable, donc plus de sommet $x \neq s \neq p$ excédentaire, donc ϕ est un flot. Donc le flot ϕ est maximum, car il n'existe pas de chemin de s à p, par théorème de Ford-Fuckerson.

2.1.5 Analyse de complexité et terminaison

Lemne 2.5 *Mêmes notations. Pour tout sommet x excédentaire, il existe un chemin de x à s dans R_ϕ.*

Preuve Posons $U = \{y \in V / \exists$ un chemin de x à y dans $R_\phi\}$. Montrons par l'absurde que $s \in U$. Supposons que $s \in U$ On sait que $\forall (y, z) \in U \times \overline{U}$, on a $\phi(z, y) = 0$ (sinon $c_\phi(y, z) = c(y, z) - \phi(y, z)$. Donc $\phi(\overline{U}, U) \leq 0$. $e(U) = \sum_{y \in U} e(y) = \sum_{y \in U} \phi(V, y) = \phi(V, U) = \phi(U, U) + \phi(\overline{U}) \leq 0$. Or, $\forall y \in V, e(y) \geqslant 0$. Donc, $\forall y \in U, e(y) = 0$, en particulier $e(x) = 0$. Absurde.

Lemne 2.6 *Mêmes notations. On a toujours $h(x) \leqslant 2|v| - 1$.*

Preuve Soit x le sommet venant d'être élevé. x est forcément excédentaire. Donc il existe $\mu = (s = x_0, x_1, ..., x_k = p)$ un chemin élémentaire dans R_ϕ. $\forall i, h(x_i) \leqslant h(x_{i+1} + 1$. Donc $h(x_0) \leqslant h(x_k) + k \leqslant 2|v| + 1$.

Corollaire 2.1 *Le nombre total d'élévations dans l'algorithme est inférieur à $2|V|^2$*

Lemne 2.7 *Il y a au plus $2|V||E|$ poussées saturantes.*

Preuve Considérons une poussée saturante sur $(x, y) \in E_\phi$. Elle fait disparaître $(x, y) \in E$. On compte pour $x, y \in E$ le nombre maximum de poussées saturantes sur (x, y) et (y, x). On remarque que de telles poussées n'existent que si $(x, y) \in E$ ou $(y, x) \in E$. Si on effectue une poussée saturante sur (x, y), (x, y) disparaît de R_ϕ et il faut une poussée sur (y, x) pour qu'il revienne. Si on effectue une poussée saturante sur (x, y), on avait $h(x) = h(y) + 1$. On en déduit qu'entre deux poussées sur (x, y), $h(y)$ augmente d'au moins 2, et réciproquement pour (y, x) et $h(x)$. On considère la valeur de $h(x) + h(y)$ au cours des poussées saturantes sur (x, y) ou (y, x). Lors de la première $h(x) + h(y) \geqslant 1$. Au cours de la dernière, $h(x) + h(y) \leqslant 2(2|V| + 1)$ d'où $h(x) + h(y) \leqslant 4|V|$. Le nombre de poussées saturantes entre x et y est donc inférieur à $\frac{4|V|}{2} = 2|V|$. En raisonnant sur l'ensembles des arrêtes, on obtient bien un maximum à $2|V||E|$.

Lemne 2.8 *Il y a au plus* $4|V|^2(|V| + |E|)$ *poussées non saturantes.*

Preuve On définit la fonction de potentiel $\Phi = \sum_{y \in S} h(x)$ où S est l'ensemble des sommets excédentaires. Au départ, $\Phi = 0$. Considérons l'évolution de Φ.

opération d'élévation Φ augmente d'au plus $2|V|$ (h ne varie pas, sauf pour un sommet de hauteur, mais hauteur bornée par $2|V|$. Pas de changements de S.

opération de poussée saturante sur (x, y). h ne varie nulle part. On a éventuellement un nouveau sommet excédentaire, y. Φ augmente d'au plus $2|V|$.

opération de poussée non saturante sur (x, y). h ne varie nulle part. On a éventuellement un nouveau sommet excédentaire, y. x n'est plus excédentaire ; disparition de x dans S. Φ diminue d'au moins $h(x)$. De plus, Φ décroit de $h(x) - h(y) = 1$. La valeur maximum que peut atteindre Φ est bornée par $2|V|\times$ nombres d'élévation $+2|V|\times$ nombres de poussées saturantes $\leqslant 4|V|^2(|V| + |E|)$. Il y en a au plus $4|V|^2(|V| + |E|)$ poussées non saturantes.

Théorème 2.2 *Avec* $m = |V|$ *et* $n = |E|$, *si* $n = O(m)$, *l'algorithme réalise au plus* $O(n^2m)$ *opérations élémentaires.*

Théorème 2.3 *L'algorithme de préflot est de complexité* $O(n^2m)$.