M1Info - Optimisation et Recherche Opérationnelle

Cours 5- Flots et Couplages

Semestre Automne 2018-2019 - Université Lyon 1

département Informatique
Faculté des Sciences et Technologies
Université Claude Bernard Lyon 1

Définition du problème

- **Entrée :** Un graphe orienté $G = (V, A)$ sans cycle avec un sommet source s et un sommet puits p, et des capacités c_{ij} (entière) sur tous les arcs.

Définition du problème

- **Entrée :** Un graphe orienté $G = (V, A)$ sans cycle avec un sommet source s et un sommet puits p, et des capacités c_{ij} (entière) sur tous les arcs.
- **Sortie :** Un flot, càd, une fonction $f : A(G) \to \mathbb{N}$ qui attribue à chaque arc (i, j) une valeur $f_{ij} \geq 0$, telle que :
  - La valeur du flot sur un arc ne dépasse pas sa capacité : $f_{ij} \leq c_{ij}$ pour tout arc (i, j).
  - Ce qui entre dans un sommet (différent de s et p) est égal à ce qui sort :

$$\forall i \in V(G) \setminus \{s, p\}, \sum_{k | ki \in A(G)} f_{ki} = \sum_{j | ij \in A(G)} f_{ij}$$

Définition du problème

- **Entrée :** Un graphe orienté $G = (V, A)$ sans cycle avec un sommet source s et un sommet puits p, et des capacités c_{ij} (entière) sur tous les arcs.
- **Sortie :** Un flot, càd, une fonction $f : A(G) \rightarrow \mathbb{N}$ qui attribue à chaque arc (i, j) une valeur $f_{ij} \geq 0$, telle que :
  - La valeur du flot sur un arc ne dépasse pas sa capacité : $f_{ij} \leq c_{ij}$ pour tout arc (i, j).
  - Ce qui entre dans un sommet (différent de s et p) est égal à ce qui sort :

$$\forall i \in V(G) \setminus \{s, p\}, \sum_{k | ki \in A(G)} f_{ki} = \sum_{j | ij \in A(G)} f_{ij}$$

- **Objectif :** Maximiser la valeur du flot, $val(f)$, càd la quantité qui sort de s : $val(f) = \sum_{si \in A(G)} f_{si}$.

Définition du problème

- **Entrée :** Un graphe orienté $G = (V, A)$ sans cycle avec un sommet source s et un sommet puits p, et des capacités c_{ij} (entière) sur tous les arcs.
- **Sortie :** Un flot, càd, une fonction $f : A(G) \to \mathbb{N}$ qui attribue à chaque arc (i, j) une valeur $f_{ij} \geq 0$, telle que :
  - La valeur du flot sur un arc ne dépasse pas sa capacité : $f_{ij} \leq c_{ij}$ pour tout arc (i, j).
  - Ce qui entre dans un sommet (différent de s et p) est égal à ce qui sort :

$$\forall i \in V(G) \setminus \{s, p\}, \sum_{k | ki \in A(G)} f_{ki} = \sum_{j | ij \in A(G)} f_{ij}$$

- **Objectif :** Maximiser la valeur du flot, $val(f)$, càd la quantité qui sort de s : $val(f) = \sum_{si \in A(G)} f_{si}$.

On notera $val(f_{max})$ la valeur optimale

Quelques définitions

- Un arc est saturé si $f_{ij} = c_{ij}$

Quelques définitions

- Un arc est saturé si $f_{ij} = c_{ij}$
- Un flot est complet si tous les chemins de s à p ont un arc saturé.

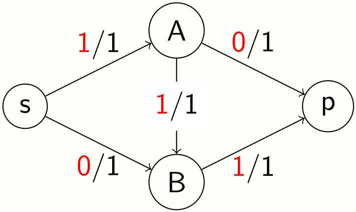
Quelques définitions

- Un arc est saturé si $f_{ij} = c_{ij}$
- Un flot est complet si tous les chemins de s à p ont un arc saturé.
- Un flot est maximal si sa valeur est optimale.

Quelques définitions

- Un arc est saturé si $f_{ij} = c_{ij}$
- Un flot est complet si tous les chemins de s à p ont un arc saturé.
- Un flot est maximal si sa valeur est optimale.

Attention ! maximal $\Rightarrow$ complet mais l'inverse n'est pas vrai :

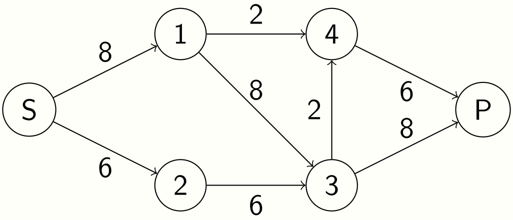


Algorithme de Ford-Fulkerson (1956)

Première étape : trouver un flot complet

a. Tant qu'il existe un chemin de s à p sans arc saturé
   - Prendre un tel chemin et calculer la plus petite valeur $c_{ij} - f_{ij}$ sur ce chemin
   - Augmenter le flot de cette valeur sur tout le chemin

Exemple



Algorithme de Ford Fulkerson - étape 2

Deuxième étape : améliorer le flot avec des chaînes augmentantes

a. Marquer les sommets de la manière suivante
   - Initialement, seul le sommet s est marqué.
   - Si uv est un arc non saturé et u est marqué, alors on marque v.
   - Si uv est un arc contenant du flot et si v est marqué, alors on marque u.
   - Continuer jusqu'à marquer p ou bien jusqu'à ne plus pouvoir marquer de sommets.

Algorithme de Ford Fulkerson - étape 2

Deuxième étape : améliorer le flot avec des chaînes augmentantes

a. Marquer les sommets de la manière suivante
   - Initialement, seul le sommet s est marqué.
   - Si uv est un arc non saturé et u est marqué, alors on marque v.
   - Si uv est un arc contenant du flot et si v est marqué, alors on marque u.
   - Continuer jusqu'à marquer p ou bien jusqu'à ne plus pouvoir marquer de sommets.

b. Si p est atteint :
   - exhiber un chemin de sommets marqués et calculer sa capacité disponible : la plus petite valeur parmi les $c_{ij} - f_{ij}$ pour les arcs marqués de i vers j et f_{ij} pour les arcs marqués de j vers i
   - Ajouter ce chemin au flot courant en retranchant la valeur pour les arcs marqués en arrière.
   - Recommencer à l'étape 2a.

Algorithme de Ford Fulkerson - étape 2

Deuxième étape : améliorer le flot avec des chaînes augmentantes

a. Marquer les sommets de la manière suivante
   - Initialement, seul le sommet s est marqué.
   - Si uv est un arc non saturé et u est marqué, alors on marque v.
   - Si uv est un arc contenant du flot et si v est marqué, alors on marque u.
   - Continuer jusqu'à marquer p ou bien jusqu'à ne plus pouvoir marquer de sommets.

b. Si p est atteint :
   - exhiber un chemin de sommets marqués et calculer sa capacité disponible : la plus petite valeur parmi les $c_{ij} - f_{ij}$ pour les arcs marqués de i vers j et f_{ij} pour les arcs marqués de j vers i
   - Ajouter ce chemin au flot courant en retranchant la valeur pour les arcs marqués en arrière.
   - Recommencer à l'étape 2a.

c. Si p n'est pas atteint, le flot est maximal.

Complexité

Quelques remarques :

- Le choix des chemins est libre... mais influence fortement.

Complexité

Quelques remarques :

- Le choix des chemins est libre... mais influence fortement.
- Si on a de la chance, l'étape 1 suffit.

Complexité

Quelques remarques :

- Le choix des chemins est libre... mais influence fortement.
- Si on a de la chance, l'étape 1 suffit.

Complexité :

- Calculer un chemin dans un graphe :

Complexité

Quelques remarques :

- Le choix des chemins est libre... mais influence fortement.
- Si on a de la chance, l'étape 1 suffit.

Complexité :

- Calculer un chemin dans un graphe : $O(m)$
- Combien de calculs de chemins ? A chaque fois, le flot augmente au moins d'une unité $\rightarrow$ au max $val(f_{max})$ chemins

Complexité

Quelques remarques :

- Le choix des chemins est libre... mais influence fortement.
- Si on a de la chance, l'étape 1 suffit.

Complexité :

- Calculer un chemin dans un graphe : $O(m)$
- Combien de calculs de chemins ? A chaque fois, le flot augmente au moins d'une unité $\rightarrow$ au max $val(f_{max})$ chemins
- Complexité finale : $O(val(f_{max}) \times m)$

Complexité

Quelques remarques :

- Le choix des chemins est libre... mais influence fortement.
- Si on a de la chance, l'étape 1 suffit.

Complexité :

- Calculer un chemin dans un graphe : $O(m)$
- Combien de calculs de chemins ? A chaque fois, le flot augmente au moins d'une unité $\rightarrow$ au max $val(f_{max})$ chemins
- Complexité finale : $O(val(f_{max}) \times m)$
- ... non polynomial !

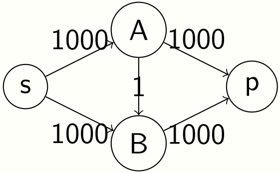
Complexité

Quelques remarques :

- Le choix des chemins est libre... mais influence fortement.
- Si on a de la chance, l'étape 1 suffit.

Complexité :

- Calculer un chemin dans un graphe : $O(m)$
- Combien de calculs de chemins ? A chaque fois, le flot augmente au moins d'une unité $\rightarrow$ au max $val(f_{max})$ chemins
- Complexité finale : $O(val(f_{max}) \times m)$
- ... non polynomial ! Et cela peut être vraiment très mauvais :



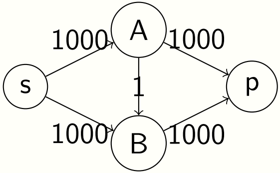
Complexité

Quelques remarques :

- Le choix des chemins est libre... mais influence fortement.
- Si on a de la chance, l'étape 1 suffit.

Complexité :

- Calculer un chemin dans un graphe : $O(m)$
- Combien de calculs de chemins ? A chaque fois, le flot augmente au moins d'une unité $\rightarrow$ au max $val(f_{max})$ chemins
- Complexité finale : $O(val(f_{max}) \times m)$
- ... non polynomial ! Et cela peut être vraiment très mauvais :



- Variantes polynomiales où l'on choisit bien les chemins

Correction de l'algo

Pourquoi le flot est-il maximal ?

- Borne supérieure naïve :

Correction de l'algo

Pourquoi le flot est-il maximal ?

- Borne supérieure naïve :
  - ce qui sort au max de s : $val(f_{max}) \leq \sum_{si \in A} c_{si}$
  - ce qui arrive au max à p : $val(f_{max}) \leq \sum_{ip \in A} c_{ip}$

Correction de l'algo

Pourquoi le flot est-il maximal ?

- Borne supérieure naïve :
  - ce qui sort au max de s : $val(f_{max}) \leq \sum_{si \in A} c_{si}$
  - ce qui arrive au max à p : $val(f_{max}) \leq \sum_{ip \in A} c_{ip}$
- Se généralise avec une coupe :

Correction de l'algo

Pourquoi le flot est-il maximal ?

- Borne supérieure naïve :
  - ce qui sort au max de s : $val(f_{max}) \leq \sum_{si \in A} c_{si}$
  - ce qui arrive au max à p : $val(f_{max}) \leq \sum_{ip \in A} c_{ip}$
- Se généralise avec une coupe :

Coupe : partition des sommets en deux parties $X, \overline{X}$, avec $s \in X$ et $p \notin X$

Correction de l'algo

Pourquoi le flot est-il maximal ?

- Borne supérieure naïve :
  - ce qui sort au max de s : $val(f_{max}) \leq \sum_{si \in A} c_{si}$
  - ce qui arrive au max à p : $val(f_{max}) \leq \sum_{ip \in A} c_{ip}$
- Se généralise avec une coupe :

Coupe : partition des sommets en deux parties $X, \overline{X}$, avec $s \in X$ et $p \notin X$

Valeur de la coupe, notée $val(X)$: somme des capacités des arcs qui sortent de X :

$$val(X) = \sum_{ij \in A, i \in X, j \notin X} c_{ij}$$

Correction de l'algo

Pourquoi le flot est-il maximal ?

- Borne supérieure naïve :
  - ce qui sort au max de s : $val(f_{max}) \leq \sum_{si \in A} c_{si}$
  - ce qui arrive au max à p : $val(f_{max}) \leq \sum_{ip \in A} c_{ip}$
- Se généralise avec une coupe :

Coupe : partition des sommets en deux parties $X, \overline{X}$, avec $s \in X$ et $p \notin X$

Valeur de la coupe, notée $val(X)$: somme des capacités des arcs qui sortent de X :

$$val(X) = \sum_{ij \in A, i \in X, j \notin X} c_{ij}$$

On a pour tout flot et toute coupe :

$$val(f) \leq val(X)$$

Un nouveau problème : trouver la coupe min

- Une coupe donne une borne supérieure sur la valeur du flot

Un nouveau problème : trouver la coupe min

- Une coupe donne une borne supérieure sur la valeur du flot
- On veut donc trouver la plus petite coupe !

Un nouveau problème : trouver la coupe min

- Une coupe donne une borne supérieure sur la valeur du flot
- On veut donc trouver la plus petite coupe !
- Nouveau problème : Coupe Minimum.

Un nouveau problème : trouver la coupe min

- Une coupe donne une borne supérieure sur la valeur du flot
- On veut donc trouver la plus petite coupe !
- Nouveau problème : COUPE MINIMUM.
- C'est le dual du problème du flot...

Un nouveau problème : trouver la coupe min

- Une coupe donne une borne supérieure sur la valeur du flot
- On veut donc trouver la plus petite coupe !
- Nouveau problème : COUPE MINIMUM.
- C'est le dual du problème du flot... et il y a dualité forte !

Un nouveau problème : trouver la coupe min

- Une coupe donne une borne supérieure sur la valeur du flot
- On veut donc trouver la plus petite coupe !
- Nouveau problème : COUPE MINIMUM.
- C'est le dual du problème du flot... et il y a dualité forte !

Théorème (flot max-coupe min) :

$$val(f_{max}) = val(coupe_{min})$$

Un nouveau problème : trouver la coupe min

- Une coupe donne une borne supérieure sur la valeur du flot
- On veut donc trouver la plus petite coupe !
- Nouveau problème : Coupe Minimum.
- C'est le dual du problème du flot... et il y a dualité forte !

Théorème (flot max-coupe min) :

$$val(f_{max}) = val(coupe_{min})$$

Preuve en utilisant Ford-Fulkerson : les sommets marqués à la fin de l'algorithme donne une coupe de la valeur du flot.

Bilan

- Pour montrer qu'un flot est maximal

Bilan

- Pour montrer qu'un flot est maximal
  - donner une coupe de même valeur
- Pour montrer qu'un flot n'est pas maximal

Bilan

- Pour montrer qu'un flot est maximal
  - donner une coupe de même valeur
- Pour montrer qu'un flot n'est pas maximal
  - Trouver une chaine augmentante
- Pour montrer qu'une coupe est minimum

Bilan

- Pour montrer qu'un flot est maximal
  - donner une coupe de même valeur
- Pour montrer qu'un flot n'est pas maximal
  - Trouver une chaine augmentante
- Pour montrer qu'une coupe est minimum
  - Trouver un flot de la même valeur.

Retour au problème du couplage

- **Données :** graphe biparti entre parties A et B.
- **Solution :** un sous-ensemble d'arêtes sans extrémités communes.
- **Objectif :** trouver un couplage de plus grande taille

Equivalent à un problème de flot :

- On ajoute s qui pointe vers A avec capacité 1 sur chaque arc
- On ajoute p vers qui tout B pointe, avec capacité 1
- Tous les arcs sont orientés de A vers B, avec capacité 1

Couplage parfait $\Leftrightarrow$ flot de taille $|A|$.

Théorème de Hall

Conditions pour un couplage parfait ?

- $|A| = |B|$

Théorème de Hall

Conditions pour un couplage parfait ?

- $|A| = |B|$
- Si $X \subseteq A$ a strict moins de $|X|$ voisins, couplage parfait impossible

Théorème de Hall

Conditions pour un couplage parfait ?

- $|A| = |B|$
- Si $X \subseteq A$ a strict moins de $|X|$ voisins, couplage parfait impossible

C'est une condition nécessaire et suffisante :

Théorème de Hall (lemme des mariages) : Si G est un graphe biparti avec $|A| = |B|$, il existe un couplage parfait ssi, pour tout $X \subseteq A$, $|X| \leq |N(X)|$.

($N(X)$ désigne l'ensemble des voisins des sommets de X)

Preuve du théorème de Hall

- S'il existe un couplage parfait (CP), alors la condition de Hall est clairement vérifiée (X a au moins les couplés comme voisins)

Preuve du théorème de Hall

- S'il existe un couplage parfait (CP), alors la condition de Hall est clairement vérifiée (X a au moins les couplés comme voisins)
- On suppose qu'il n'existe pas de CP. Alors flot max $< |A|$. On applique Ford Fulkerson et on regarde le marquage final.

Preuve du théorème de Hall

- S'il existe un couplage parfait (CP), alors la condition de Hall est clairement vérifiée (X a au moins les couplés comme voisins)
- On suppose qu'il n'existe pas de CP. Alors flot max $< |A|$. On applique Ford Fulkerson et on regarde le marquage final.
- s est marqué et au moins un sommet de A. Soit X_A les sommets de A marqués et X_B les sommets de B marqués.

Preuve du théorème de Hall

- S'il existe un couplage parfait (CP), alors la condition de Hall est clairement vérifiée (X a au moins les couplés comme voisins)
- On suppose qu'il n'existe pas de CP. Alors flot max $< |A|$. On applique Ford Fulkerson et on regarde le marquage final.
- s est marqué et au moins un sommet de A. Soit X_A les sommets de A marqués et X_B les sommets de B marqués.
- On peut montrer que $N(X_A) = X_B$.

Preuve du théorème de Hall

- S'il existe un couplage parfait (CP), alors la condition de Hall est clairement vérifiée (X a au moins les couplés comme voisins)
- On suppose qu'il n'existe pas de CP. Alors flot max $< |A|$. On applique Ford Fulkerson et on regarde le marquage final.
- s est marqué et au moins un sommet de A. Soit X_A les sommets de A marqués et X_B les sommets de B marqués.
- On peut montrer que $N(X_A) = X_B$.
- Valeur de la coup $=$ nb arcs sortants $= |A| - |X_A| + |X_B|$

Preuve du théorème de Hall

- S'il existe un couplage parfait (CP), alors la condition de Hall est clairement vérifiée (X a au moins les couplés comme voisins)
- On suppose qu'il n'existe pas de CP. Alors flot max $< |A|$. On applique Ford Fulkerson et on regarde le marquage final.
- s est marqué et au moins un sommet de A. Soit X_A les sommets de A marqués et X_B les sommets de B marqués.
- On peut montrer que $N(X_A) = X_B$.
- Valeur de la coup = nb arcs sortants $= |A| - |X_A| + |X_B|$
- pas de CP donc valeur $< |A|$. Donc $|X_B| = N(X_A) < |X_A|$. Condition de Hall non vérifiée.

Preuve du théorème de Hall

- S'il existe un couplage parfait (CP), alors la condition de Hall est clairement vérifiée (X a au moins les couplés comme voisins)
- On suppose qu'il n'existe pas de CP. Alors flot max $< |A|$. On applique Ford Fulkerson et on regarde le marquage final.
- s est marqué et au moins un sommet de A. Soit X_A les sommets de A marqués et X_B les sommets de B marqués.
- On peut montrer que $N(X_A) = X_B$.
- Valeur de la coup $=$ nb arcs sortants $= |A| - |X_A| + |X_B|$
- pas de CP donc valeur $< |A|$. Donc $|X_B| = N(X_A) < |X_A|$. Condition de Hall non vérifiée.

Remarque : Ford Fulkerson nous exhibe l'ensemble problématique !