# Recherche Opérationnelle
# CTD 2

Sandra U. Ngueveu

ENSEEIHT - LAAS
sandra.ngueveu@enseeiht.fr - ngueveu@laas.fr
http://homepages.laas.fr/sungueve

2016/2017





## Rappels de PLNE

voir CTD 1 : PLNE1.pdf

# Graphes et calculs de chemins

Graphes : Définitions, concepts et notations GraphesDefinitions.pdf





## Définitions

Graphe non orienté G=(X,A)

Graphe orienté G=(X,A)

Graphe (non) orienté valué G=(X,A,C)

Exemples et remarques





## Concepts élémentaires

Dans un graphe orienté :

- Chemin
- Circuit
- Boucle

Dans un graphe non orienté :

- Chaîne
- Cycle
- Boucle

Elémentaire/Hamiltonien, Simple/Eulérien





## Concepts élémentaires

Graphe complet

Autres types particuliers :

- graphe partiel / sous-graphe
- graphe connexe / fortement connexe
- graphe symétrique / antisymétrique
- arbre / arborescence





## Modélisation/Représentation informatique

Matrice d'incidence

Matrice d'adjacence

Liste des successeurs

## Calculs de chemins dans un graphe

### Recherche de chemins de coût min dans des graphes particuliers

- sans circuit : décomposition par niveau + parcours
- graphe à longueurs positives : Diksjtra, A*, ...

### Recherche de chemin de coût min sur graphe quelconque : **Bellmann-Ford**

- Données
- Principes
- Illustration





## Flots et coupes

Problèmes de flots dans les graphes FlotsMaxEtFlotdeCoutMax.pdf

## Unimodularité

**Définition**
Une matrice entière A : $m \times n$ de rang $m$ est unimodulaire si toute sous-matrice carrée $m \times m$ a un déterminant valant $+1$, 0 ou -1.

**Propriété**
Les propriétés suivantes sont équivalentes :

- $A$ est unimodulaire.
- le système $Ax = b, x \geq 0$, a des solutions de base entières pour tout vecteur b entier
- pour toute matrice de base $B$ de $A$, $B^{-1}$ est entière.

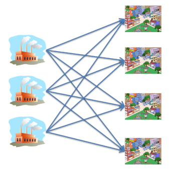
**Conséquence**
Un PL{ $\min c.x, A.x = b, x \geq 0$ } avec $A$ unimodulaire et $b$ entier a des solutions optimales entières.



## Unimodularité

**Exemple : Le problème de transport**
Il s'agit de calculer un plan de transport pour acheminer un produit de $m$ sources potentielles à $n$ destinations qui en ont besoin. Chaque source a une disponibilité $a_i$ et et chaque destination a une demande minimale $b_j$.
Le coût unitaire de transport de $i$ à $j$ est $c_{ij}$. On note $x_{ij}$ la quantité à faire transiter de $i$ à $j$. L'objectif est de minimiser le coût total.

... MODELE MATH ... ?






## Plan de ce cours







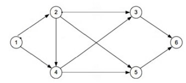
## Graphes : Rappel

Graphe = Mode de représentation/modélisation pour représenter des entités et les relations entre ces entités par des nœuds et des arcs/arêtes respectivement.

- peut représenter réseau de transport, un système, les états d'un système, ...
- peut être orienté/non orienté, valué/non valué, complet/non complet

**Notation** : Graphe orienté valué non complet $G = (X, U, C)$

- ensemble $X$ de $n$ nœuds (ou sommets)
- ensemble $U$ de $m$ arcs (couples $(x, y)$ de $X$ ).
- valeur $C_{ij}$ affectée à chaque arc $(i, j)$.








## Flots

Flot

- modélise la circulation de flux/fluides entre un noeud-source $s$ et un noeud-destination $t$

Modèle de graphe orienté valué utilisé $G = (X, U, C, s, t)$ :

- $X$ ensemble de $n$ nœuds, $U$ ensemble de $m$ arcs
- $C_{ij}$ capacité maximale (entière) de l'arc $(i, j)$
- nœud source $s$ et nœud puits $t$.

Un flot est une application $\phi$ de $U$ dans $N$ :

- respectant les capacités des arcs
- $r$ conservant le flot en chaque noeud (Kirchhoff)
- de débit $F$ égal au flot sortant de $s$ ou entrant en $t$





## Flot maximal

Le problème du flot maximal consiste à faire circuler un flot de débit maximal entre deux noeuds donnés $s$ et $t$.

Modèle mathématique

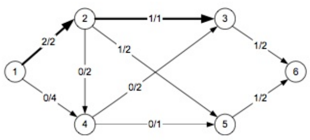




## Flot maximal

Le problème du flot maximal consiste à faire circuler un flot de débit maximal entre deux noeuds donnés $s$ et $t$.

Exemple : $s = 1$, $t = 6$, flot de débit 2 (flux/capacité). Arcs saturés : trait double ou épais. Flot non optimal.








## Modélisation de cas particuliers

Graphe non orienté

Sources multiples et disponibilités limitées

Capacité sur un noeud (s'il y a une limite sur la capacité de traitement au sein d'une des entités)

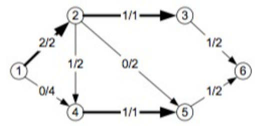
Noeud $j$ avec un seul prédécesseur $i$ et un seul successeur $k$

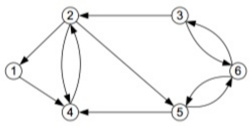
## Calcul du flot maximal

Un chemin améliorant va de $s$ à $t$ avec des arcs $(i,j)$ insaturés : $\phi ij < Cij$.

Un flot est complet si tout chemin de $s$ à $t$ comporte au moins un arc saturé. Attention, il n'est pas forcément optimal !

Exemple : le flot peut augmenter sur la chaîne améliorante [1,4,2,5,6] qui emprunte l'arc (2,4) à l'envers.








## Algorithme de Ford-Fulkerson

Pour décrire les augmentations possibles, on peut construire un graphe d'écart (ou graphe résiduel) $G_e(\phi)$. Il contient :

- les noeuds de $G$
- les arcs non saturés,
- un arc $(j,i)$ pour tout arc $(i,j)$ de flot non nul dans $G$.

Une chaîne améliorante de $G$ correspond à un chemin de $s$ à $t$ dans $G_e(\phi)$.





## Algorithme de Ford-Fulkerson

Principe

Procédure de marquage

Complexité : Implémentable en $O(n.m^2)$. Il existe des algorithmes plus rapides mais plus compliqués :

- celui de Karzanov en $O(n^3)$
- celui d'Ahuja en $O(m.n.\log C_{max})$





## Coupes

Une coupe (cut) de $G$ est une partition des noeuds $Z = (S,T)$ avec $s$ dans $S$ et $t$ dans $T$.

Un arc sortant de $Z$ va de $S$ à $T$, un arc entrant va de $T$ à $S$.
La capacité de la coupe est la somme des capacités de ses arcs sortants : $C(S,T) = \sum_{i \in S, j \in T, (i,j) \in U} C_{ij}$.

**Propriété**
Le débit de tout flot est inférieur ou égal à la capacité de toute coupe.





## Flot maximal / Coupe minimale

**Théorème max-flow, min-cut**

- Le débit maximal d'un flot de G est égal à la capacité minimale des coupes de G.
- L'algorithme de Ford et Fulkerson trouve un flot maximal de s à t.

Un flot est donc maximal si on trouve une coupe de capacité égale à son débit.





## Flot de coût minimal

Montrer que le problème de coupe minimale est le dual du problème de flot maximum.

| Le pb de flot maximum | Le pb de coupe minimale |
| --- | --- |
| ... MODELE MATH + vars duales... ? | ... MODELE MATH ... ? |





## Flot de coût minimal

1 Flot maximum / Coupe minimale

2 Flots de coût minimal





## Flot de coût minimal

Soit un réseau $G = (X, U, C, W, s, t)$ avec :

- $X$ ensemble de $n$ nœuds, $U$ ensemble de $m$ arcs,
- $C_{ij}$ capacité (entière) de l'arc $(i,j)$,
- $W_{ij}$ coût de passage d'une unité de flux sur $(i,j)$,
- une source $s$ et un puits $t$.

Le problème du flot de coût minimal consiste à faire circuler un flot de $s$ à $t$, de débit donné **F**, en minimisant le coût total.

Modèle mathématique

## Flot maximal de coût minimal

Comme pour le flot max, il existe des algorithmes de graphes plus rapides que la PL, se basant sur graphe d'écart (ou graphe résiduel). Pour un flot $\phi$, le graphe d'écart $G_e(\phi)$ doit tenir compte des coûts

- Un arc non saturé $(i,j)$ de $G$ est conservé dans le graphe d'écart avec son coût
- tout arc $(i,j)$ de flot non nul de $G$ donne lieu à un arc inversé $(j,i)$ de coût $-W_{ij}$ dans $G_e(\phi)$
- Comme pour le flot maximal, un chemin $\pi$ de s à t dans le graphe d'écart correspond à une chaîne améliorante $\mu$ de $G$ avec des arcs avant et arrière.

Une augmentation de flot le long de $\mu$ donnera un gain ou une perte selon le coût total du chemin $\pi$.





## Algorithme de Busacker et Gowen

**Principe**

- On part du flot nul : débit $Q = 0$ et coût total $CT = 0$.
- On calcule un flot de débit $F$ imposé, et de coût minimal.
- **Différence avec Ford-Fulkerson** : les augmentations se font le long de chaînes améliorantes de coût minimal.
- A chaque itération, on cherche donc un chemin $\pi$ de coût minimal $z$, de $s$ à $t$ dans le graphe d'écart.





## Algorithme de Busacker et Gowen

1. Il faut utiliser l'algo de Bellman pour calculer $\pi$, à cause des coûts négatifs possibles dans $G_e(\phi)$.
2. Si $F = \mathrm{constante}$, l'algo se termine quand $Q = F$ grâce au plafonnement de $\delta$. Si $F = \infty$, il stoppe quand il ne trouve plus de chaîne améliorante : on a alors un flot maximal et de coût minimal.

Dans les 2 cas, le coût $CT$ est minimal.

L'optimalité de l'algo repose sur la propriété suivante, qu'on admettra :

- A chaque itération, le flot obtenu est de coût minimal parmi tous les flots de débit $Q$.

**Complexité**

- Soit $K$ le maximum des capacités. L'algorithme de Busacker et Gowen est en $O(m.n^2.K)$, mais il est rapide en moyenne.