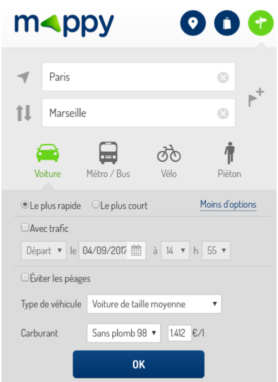
Exemple 1 : cheminer



Critères

1. Le temps
2. La distance
3. Le coût

Calcul du chemin minimisant un critère

Solution trouvée facilement par un algorithme de graphes

Prédécesseurs et successeurs

Définition - **Successeur d'un sommet**

$\Gamma(v) = \{\text{successeurs du sommet v}\}$

$V \mapsto P(V)$ (aussi noté Γ^+)

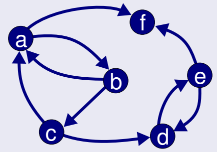
Définition - **Prédécesseur d'un sommet**

$\Gamma^{-1}(v) = \{\text{prédécesseurs du sommet v}\}$

Aussi noté Γ^-

Exemple

- $\Gamma(b) = \{a, c\}$
- $\Gamma(f) = \emptyset$
- $\Gamma^{-1}(b) = \{a\}$
- $\Gamma^{-1}(d) = \{c, e\}$



Chemin et circuit

Définition - **Chemin**

Suite d'arcs telle que l'extrémité terminale d'un arc coïncide avec l'extrémité initiale de l'arc suivant

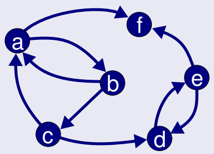
Définition - **Circuit**

Chemin dont les deux extrémités coïncident

- chemin **simple** : pas deux fois le même arc
- chemin **élémentaire** : pas deux fois le même sommet

Exemple

- Chemin :
  - (ab,bc,cd)
  - (de, ef)
- Circuit :
  - (a,b,a) ou (ou (ab, ba))
  - (a,b,c,a) (ou (ab,bc,ca))



Racine, degrés

Définition - **Racine**

Sommet r tel qu'il existe un chemin de r à tout autre sommet du graphe

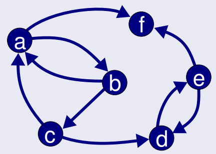
Définition - **Degré intérieur** (resp. **extérieur**) d'un sommet x

Nombre d'arcs dont x est l'extrémité terminale (noté $d^-(x)$)

resp. initiale resp. $d^+(x)$

Exemples

- Racine : a
- $d^-(a) = 2$, $d^-(f) = 2$
- $d^+(a) = 2$, $d^+(b) = 2$, $d^+(f) = 0$



Graphes non orientés

Définition - Arête

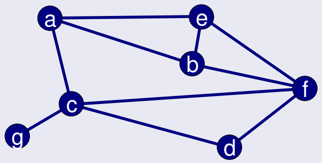
Arc "sans orientation"

Notation - Graphe non orienté

$$\underset{\text{Ensemble de sommets}}{G = (V}, \underset{\text{Ensemble d'arêtes}}{E)}$$

Exemple

- $V = \{a, b, c, d, e, f, g\}$
- $E = \{[ae], [ab], [ef], ...\}$

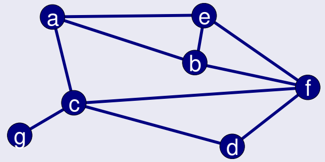


Définition - Chaîne

Séquence d'arêtes telle que toute arête est adjacente à l'arête qui la suit et à celle qui la précède

Exemple

- Chaîne : $[a, e, f, d]$ (ou $\{[ae], [ef], [fd]\}$)



Définition - Voisinage

Les sommets x et y sont dits voisins si $[xy] \in E$

Notation - $N(x)$

$N(x) = \{\text{voisins de } x\}$

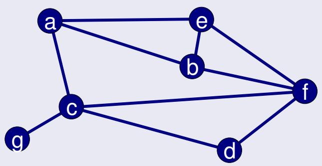
Définition - Degré

$d(x) = |N(x)|$

Nombre d'arêtes adjacentes à x

Exemple

- b est voisin de a et e
- $N(c) = \{a, d, f, g\}$
- $d(b) = 3$
- $d(c) = 4$



Cycle élémentaire

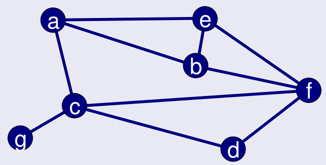
Définition - Cycle (élémentaire)

Chaîne dont les deux extrémités coïncident

(et qui ne passe pas 2 fois par le même sommet)

Exemple

- Cycle : $[aefba]$



Définition - Cycle Hamiltonien

Cycle élémentaire passant par tous les sommets

Définition - **Relation de connexité $\mathcal{R}$**

Soit x et y deux sommets d'un graphe $G = (V, A)$

- $x\mathcal{R}y \Leftrightarrow x$ et y sont reliés par une chaîne

Définition - **Composante connexe**

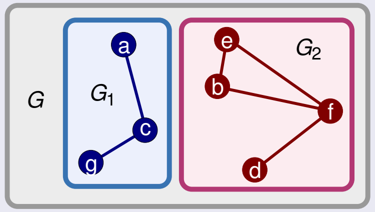
$\mathcal{R}$ est une relation d'équivalence dont les classes d'équivalences sont appelées **composantes connexes**

Définition - **Graphe connexe $G = (V, A)$**

G ne possède qu'une unique composante connexe

Exemple

- $a\mathcal{R}g$
- G_1 et G_2 connexes
- G non connexe



Arbre

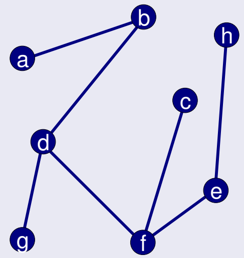
Définition - **Arbre**

Graphe connexe et sans cycle

Définition - **Forêt**

Graphe sans cycle

Exemple



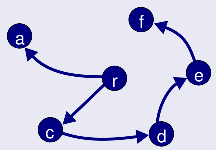
Graphes orientés - Arborescence

Définition - **Arborescence**

$G = (V, A)$ arbre possédant une racine r telle que

- r est reliée à tout $v \in V$ par un chemin unique

Exemple



Propriété

- $d^-(r) = 0$
- $d^-(x) = 1$ pour tout $v \neq r$

arborescence = "arbre enraciné" = "arbre" en informatique

- arbre généalogique, tournois, arbre des espèces animales,...

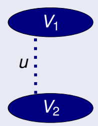
Preuve d'optimalite - Algorithme de Kruskal

Notations

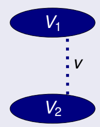
- H : arbre obtenu par l'algorithme de Kruskal de poids $p(H)$
- $H^{(1)}$: arbre optimal de poids $p(H^{(1)})$
- $u \in H^{(1)} \backslash H$ reliant V_1 et V_2
  Avec $V_1 \cup V_2 = V$
- $v \in H \backslash H^{(1)}$ reliant V_1 et V_2

Montrons que $p(H) = p(H^{(1)})$

Optimal - $H^{(1)}$



Algorithme - H



- $p(u) \geq p(v)$ (propriété 1)
- $p(v) \geq p(u)$ ($H^{(1)}$ est optimal)

Soit $H^{(2)} = H^{(1)} \cup \{v\} \backslash \{u\}$
On a donc $p(H^{(2)}) = p(H^{(1)})$

On répète le processus...

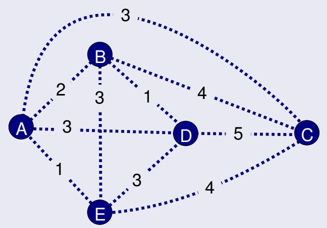
Considérons $w \in H^{(2)} \backslash H$
On a donc $p(H^{(3)}) = p(H^{(1)})$

On répète jusqu'à ce que $p(H^{(\dots)}) = p(H)$

Algorithme glouton pour le voyageur de commerce

Données : $G = (V, E, p)$: graphe initial
Résultat : $H(V, E_2)$: cycle hamiltonien
$k \leftarrow 0$
$E_2 \leftarrow \emptyset$
$L \leftarrow$ Liste des arêtes de G triées par ordre de longueur croissante
pour k allant de 1 à n **faire**
  $w \leftarrow$ 1ère arête de L ne formant pas de sous-cycle avec E_2 et telle que les degrés des sommets restent ≤ 2
  $E_2 \leftarrow E_2 \cup \{w\}$
retourner $H = (V, E_2)$

Exemple

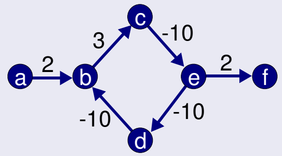


$L = (AE, BD, AB, BE, AD, DE, AC, BC, CE, CD)$

Solution heuristique de valeur 13

Définition - Circuit absorbant

Circuit de longueur négative



Théorème

- Il existe un chemin de longueur minimale finie de r à tous les sommets du graphe

si et seulement si

- r est une racine du graphe et le graphe ne contient pas de circuit absorbant

Cas où l'on est sûr de l'absence de circuit absorbant

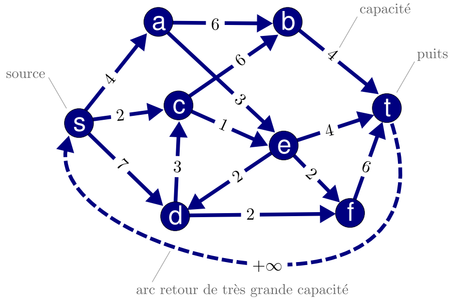
- Toutes les longueurs sont positives ou nulles
- Le graphe est sans circuit

Réseau de transport

Graphe $G = (V, A, C)$ orienté tel qu'il existe : (Capacité ≥ 0 des arcs)

- $s \in V$ une **source** ($\Gamma^-(s) = \emptyset$)
- $t \in V$ un **puits** ($\Gamma^+(t) = \emptyset$)

On ajoute $(ts) \in A$ un arc fictif **de retour**



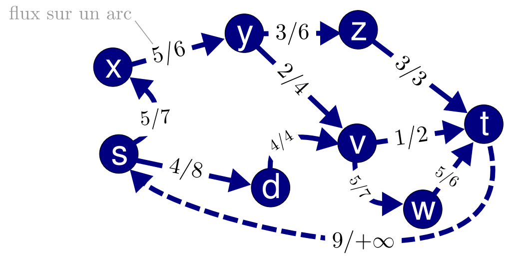
Problème de flot maximal

Comment transférer une quantité maximale de « matière » de s à t sans dépasser la capacité de chaque arc ?

Hypothèses

A chaque noeud :

- il y a conservation de la matière
- il est possible de décomposer/recomposer la matière transférée



Applications

- Réseaux routiers
- Distribution d'eau
- Réseau internet
- ...

Flot sur un réseau de transport

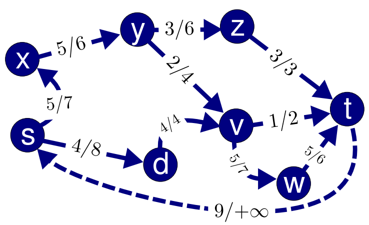
Remarque

La loi de conservation est vérifiée en s et t grâce à l'arc de retour

Défintion - **Flot complet** φ

φ est dit **complet** si et seulement si

- tout chemin de s à t comporte au moins un arc saturé

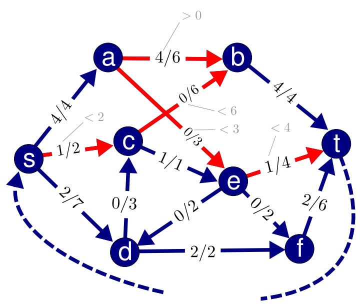


Chaîne améliorante

Définition - **Chaîne améliorante** μ pour un flot φ

Chaîne de s à t vérifiant que :

- $\varphi_{ij} < cij$ pour tout arc (ij) de μ dans le "bon sens" (de s vers t)
- $\varphi_{ij} > 0$ pour tout arc (ij) de μ dans le "mauvais sens" (de t vers s)

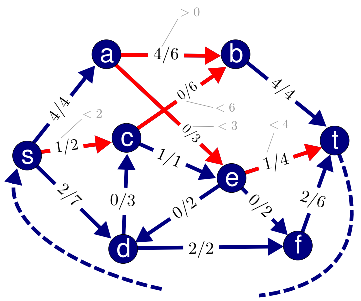


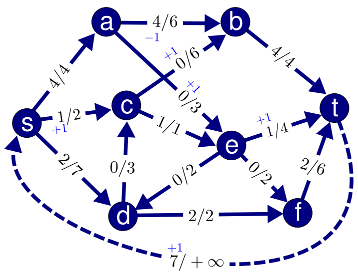
Chaîne améliorante

Définition - Chaîne améliorante μ **pour un flot** φ

Chaîne de s à t vérifiant que :

- $\underset{\ldots}{\varphi_{ij} < cij}$ pour tout arc (ij) de μ dans le "bon sens"
  de s vers t
- $\underset{\ldots}{\varphi_{ij} > 0}$ pour tout arc (ij) de μ dans le "mauvais sens"
  de t vers s



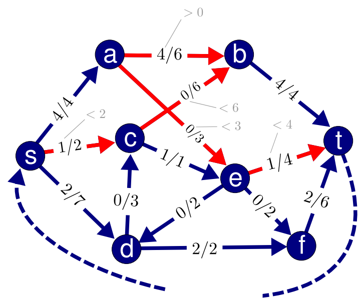


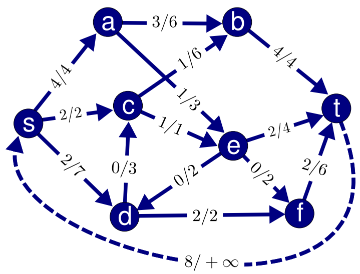
Chaîne améliorante

Définition - Chaîne améliorante μ **pour un flot** φ

Chaîne de s à t vérifiant que :

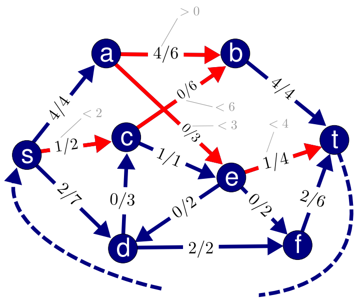
- $\underset{\ldots}{\varphi_{ij} < cij}$ pour tout arc (ij) de μ dans le "bon sens"
  de s vers t
- $\underset{\ldots}{\varphi_{ij} > 0}$ pour tout arc (ij) de μ dans le "mauvais sens"
  de t vers s





Chaîne améliorante

Soit μ une chaîne améliorante



Notations

- μ^+ = arcs de μ dans le bon sens
- μ^- = arcs de μ dans le mauvais sens

Augmentation de la valeur du flot de α

$$\alpha = \underset{\ldots}{min[min_{(ij)\in\mu^+}(cij - \varphi_{ij}), min_{(ij)\in\mu^-}\varphi_{ij}]}$$

- dans μ^+ : on augmente les flux de α
- dans μ^- : on diminue les flux de α

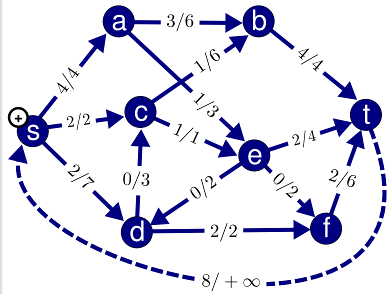
L'algorithme de Ford-Fulkerson : une procédure de marquage

Algorithme

Données : $G = (V, A, C)$
Établir un flot admissible (complet de préférence)
répéter
  Retirer toutes les marques
  Marquer '+' le sommet s
  répéter
    Marquer '$+i$' le sommet terminal j de tout arc (ij) tel que :
    - i est marqué
    - j est non marqué
    - (ij) non saturé

    Marquer '$-j$' le sommet initial i de tout arc (ij) tel que :
    - i est non marqué
    - j est marqué
    - (ij) a un flux non nul

  tant que un nouveau sommet a été marqué **et** t n'est pas marqué
  si t est marqué **alors**
    Améliorer le flux via une chaîne améliorante
tant que t est marqué



Chaîne améliorante μ

$$s \xrightarrow{5} d \xrightarrow{3} c \xrightarrow{5} b \xleftarrow{3} a \xrightarrow{2} e \xrightarrow{2} t$$

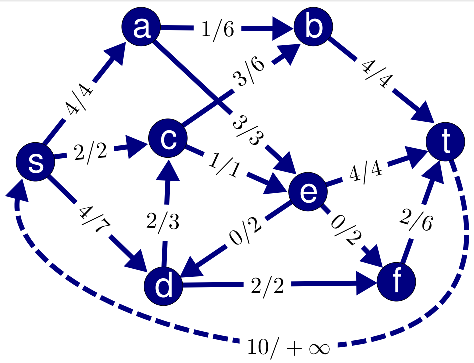
- Amélioration de $\underset{\ldots}{+2}$

Un réseau de transport

Le flot obtenu est-il optimal ?

1. Oui
2. Non

Oui, le marquage donne (+, s), (d, +s), (c, +d), (b, +c), (a, -b)



Preuve de l'algorithme

Problème de coupe minimale

Comment séparer s de t en supprimant un ensemble d'arcs de valeur totale minimale ?

"Séparer" signifie qu'il n'existe plus de chemin de s à t après suppression des arcs

Définition - Coupe (S,T)

Partition de V en deux sous-ensembles S et T telle que

- $s \in S$
- $t \in T$

Notations

- $\omega^-(T)$ = arcs entrant dans T
  $\{(i,j) \in A \mid i \in S, j \in T\}$
- $\omega^+(T)$ = arcs sortant de T
  $\{(i,j) \in A \mid i \in T, j \in S\}$

Remarque

Par définition $(ts) \notin \omega^+(T)$

Définition - Capacité d'une coupe (S,T)

$$c(S,T) = \sum_{(ij) \in \omega^-(T)} c_{ij}$$

Relation flots / coupes

Propriété

Soit $G = (V, A)$ un réseau de transport

- $\forall \varphi$ flot admissible sur G
- $\forall (S, T)$ coupe de G

On a

$$v(\varphi) \leq c(S,T)$$

Preuve

- $\sum_{(ij) \in \omega^-(T)} \varphi_{ij} = \sum_{(ij) \in \omega^+(T)} \varphi_{ij} + \varphi_{ts}$ (loi de conservation)
- On sait que $\varphi_{ts} = v(\varphi)$
- $v(\varphi) = \sum_{(ij) \in \omega^-(T)} \varphi_{ij} - \sum_{(ij) \in \omega^+(T)} \varphi_{ij} \leq \sum_{(ij) \in \omega^-(T)} c_{ij} = c(S,T)$ (flux $\leq$ capacité)

Théorème de Ford-Fulkerson

Théorème - Ford-Fulkerson, 1962

La valeur d'un flot maximal est égale à la capacité d'une coupe minimale

Propriété - CNS d'optimalité

Un flot φ de s à t est maximal si et seulement si il n'existe pas de chaîne améliorante de s à t

Preuve du théorème et de l'algorithme de Ford-Fulkerson

Notations

- φ^* : flot obtenu par l'algorithme
- S^* : ensemble des sommets marqués à la fin de l'algorithme
- T^* : ensemble des sommets non marqués à la fin de l'algorithme

Rappels

- $v(\varphi^*) = \varphi^*(t,s)$
- $(t,s) \notin \omega^+(T)$

Preuve

- Toute coupe (S,T) et tout flot φ vérifient : $v(\varphi) \leq c(S,T)$
- $v(\varphi^*) = \sum_{(ij)\in\omega^-(T^*)} \varphi^*_{ij} - \sum_{(ij)\in\omega^+(T^*)} \varphi^*_{ij}$ (loi de conservation des flux)

  $= \sum_{(ij)\in\omega^-(T^*)} c^*_{ij} - 0$ (principe de marquage)

  $= c(S^*, T^*)$
- Donc φ^* est un flot maximal et (S^*, T^*) est une coupe minimale

Passage de la forme générale à la forme standard 1/2

Il est toujours possible de mettre un PL sous forme standard

1/4 Contrainte $\leq$

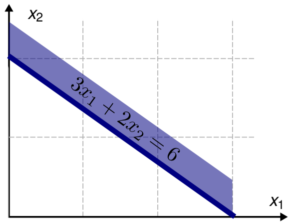
$$2x_1 + 3x_2 \leq 6 \rightarrow \begin{cases} \underbrace{2x_1 + 3x_2 + s = 6}_{} \\ s \geq 0 \end{cases}$$

Variable d'écart

2/4 Contrainte $\geq$

$$3x_1 + 2x_2 \geq 18 \rightarrow \begin{cases} 3x_1 + 2x_2 - s = 18 \\ s \geq 0 \end{cases}$$

Variable d'écart



Passage de la forme générale à la forme standard 2/2

3/4 Variable de signe quelconque

$$x \in \mathbb{R} \rightarrow \begin{cases} x = x_1 - x_2 \\ x_1, x_2 \geq 0 \end{cases}$$

4/4 Minimisation

$$\min 2x_1 - 3x_2 \rightarrow \max -2x_1 + 3x_2$$

$\min f = -\max(-f)$

Solution associée à une base

Réorganisation

- $A = (B\ N)$
- $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$ $\quad \underbrace{Bx_B + Nx_N}_{=Ax} = b$

Propriété

B inversible donc

- $x_B = B^{-1}b - B^{-1}Nx_N$

Définition - Solution associée à une base B

- $x_B = B^{-1}b$
- $x_N = 0$

Définition - Base réalisable

Base dont la solution associée est réalisable

$\Leftrightarrow B^{-1}b \geq 0$ (sinon on aurait une ou plusieurs variables négatives)

Définition - Base réalisable dégénérée

Base dont la solution réalisable comporte une variable de base nulle

$\exists b \in B \;\; x_b = 0$

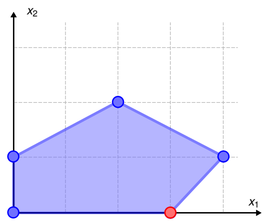
Exemple

$$\begin{array}{llllllll} \max & 2x_1 + & x_2 & & & & & \\ \text{s.c.} & x_1 - & x_2 + & x_3 & & & = & 3 \\ & x_1 + & 2x_2 & & + x_4 & & = & 6 \\ & -x_1 + & 2x_2 & & & + x_5 & = & 2 \\ & x_1, & x_2, & x_3, & x_4, & x_5 & \geq & 0 \end{array}$$

Décomposition de $Bx_B + Nx_N = b$

Considérons la base $\{x_1, x_4, x_5\}$:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_4 \\ x_5 \end{pmatrix} + \begin{pmatrix} -1 & 1 \\ 2 & 0 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 6 \\ 2 \end{pmatrix}$$



Solution de base réalisable $x_B = B^{-1}b$ et $x_N = 0$

$$\begin{pmatrix} x_1 \\ x_4 \\ x_5 \end{pmatrix} = B^{-1}b = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 6 \\ 2 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 5 \end{pmatrix}$$

Remarques

- $\{x_1, x_4, x_5\}$ est une base car la sous-matrice B associée est inversible
- $\{x_1, x_4, x_5\}$ est une base réalisable car la solution associée $(3, 0, 0, 3, 5)$ ne comporte que des valeurs positives

Théorème 2

L'optimum d'une fonction linéaire sur un polytope convexe est atteint en au moins un point extrême

Preuve

Supposons qu'aucun point extrême $\{x^1, \dots, x^p\}$ ne soit une solution optimale.
Soit x^* une solution optimale. x^* peut s'écrire comme combinaison convexe des points extrêmes : $x^* = \sum_{i=1}^{p} \lambda_i x^i$ (avec $\sum_{i=1}^{p} \lambda_i = 1$).
$cx^* < cx^i$ (cas de la minimisation)

$$\underbrace{\sum_{i=1}^{p} \lambda_i}_{=1} cx^* < \sum_{i=1}^{p} \lambda_i cx^i$$

$$cx^* < c \underbrace{\sum_{i=1}^{p} \lambda_i x^i}_{=x^*}$$

Comment savoir si une base réalisable B est optimale ?

Nécessite de réécrire l'objectif

$x_B = B^{-1}b - B^{-1}Nx_N$

Valeur de la solution associée à B

$$z = cx = c_B x_B + c_N x_N = \boxed{c_B B^{-1} b} + \boxed{(c_N - c_B B^{-1} N)}\, x_N$$

Δ_N : **coûts réduits** des variables hors base x_N

Théorème 3 (cas de la maximisation)

Une base réalisable non dégénérée B est une base optimale si et seulement si

$$\Delta_N \leq 0$$

$\Delta_N \geq 0$ en cas de minimisation

Idée de preuve

$z = c_B B^{-1} b + \Delta_N x_N$
Si $\exists j \in N \; \Delta_j > 0$ augmenter x_j de δ augmentera la valeur de z de $\Delta_j \delta$

Problématique 1

Quelle direction choisir ? (i.e., quelle variable faire entrer dans la base ?)

Corollaire du théorème 3 (cas de la maximisation)

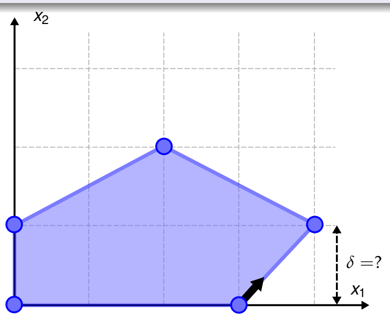
Si la base est non dégénérée et qu'$\exists$ un coût réduit > 0 alors on peut faire croître z, sinon le max est atteint

Exemple - Choix de la direction

$\Delta_2 = 3 > 0$ donc augmenter x_2 permet d'améliorer l'objectif
Si x_2 augmente de δ, z augmentera de $\Delta_2 \delta = 3\delta$

Problématique 2

Jusqu'où effectuer le déplacement ?
Les contraintes doivent être respectées



Problématique 2

Jusqu'où effectuer le déplacement ?

Les contraintes du problème indiquent de combien la variable entrant en base peut augmenter

Nécessite de reformuler les contraintes sous **forme canonique**

$$Ax = b$$
$$Bx_B + Nx_N = b$$
$$x_B = \underbrace{B^{-1}b}_{(\bar{b})} - \underbrace{B^{-1}N}_{(\bar{a})} x_N \qquad (1)$$

Quelles contraintes limitent l'entrée en base de x_2 ?

Exemple

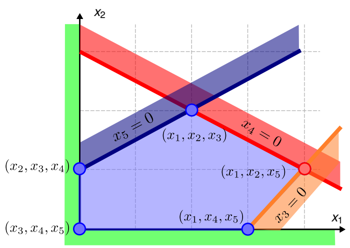
$$\overbrace{\begin{pmatrix} x_1 \\ x_4 \\ x_5 \end{pmatrix}}^{x_B} = \overbrace{\begin{pmatrix} 3 \\ 3 \\ 5 \end{pmatrix}}^{\bar{b}} - \overbrace{\begin{pmatrix} -1 & 1 \\ 3 & -1 \\ 1 & 1 \end{pmatrix}}^{\bar{a}} \overbrace{\begin{pmatrix} x_2 \\ x_3 \end{pmatrix}}^{x_N}$$

- x_3 reste hors base ($x_3 = 0$) donc

$$x_1 = 3 + x_2 \quad \geq 0 \Rightarrow x_2 \geq -3$$
$$x_4 = 3 - 3x_2 \geq 0 \Rightarrow x_2 \leq 1$$
$$x_5 = 5 - x_2 \quad \geq 0 \Rightarrow x_2 \leq 5$$

$\rightarrow$ x_2 peut être augmenté jusqu'à $\min(1, 5)$

- Si $x_2 \leftarrow 1$ alors $x_4 \leftarrow 0$
  x_4 sort donc de la base



Nouvelle solution obtenue

- En base : $x_2 = 1$, $x_1 = x_5 = 4$
- Hors base : $x_3 = x_4 = 0$

Changement de base

Remarque

Dantzig a proposé 2 critères pour déterminer

1. La variable qui entre dans la base
   Plus grand $\Delta_j > 0$
2. La variable qui sort de la base
   Plus petit rapport > 0

QCM

- A. Seul le critère 1 est impératif
- B. Seul le critère 2 est impératif
- C. Les deux sont impératifs

Reponse B
1 revient à choisir une direction (plusieurs possibles)
2 revient à voir jusqu'où on se déplace dans la direction choisie (limite fixée)

QCM

x_e doit entrer en base et tous les rapports $\frac{\bar{b}_i}{\bar{a}_{ie}}$ sont < 0.
Que peut-on en déduire ?

1. Le système n'a pas de solution
2. Le système a une solution infinie
3. On est à l'optimum

Reponse 2 : le ratio détermine la valeur limitante que prendra x_e. Si tous les ratios sont négatifs, x_e peut augmenter vers l'infini.
En effet on a $\bar{b} - \bar{a}x \geq 0$
Ce qui donne $x \leq \frac{\bar{b}}{\bar{a}}$ si $\bar{a} > 0$ et $x \geq \frac{\bar{b}}{\bar{a}}$ si $\bar{a} < 0$

Dualité faible

Théorème - Dualité faible

Pour toute solution réalisable

- x du problème primal ; et
- u du problème dual,

on a

$$\underbrace{\sum_{j=1}^{n} c_j x_j}_{\text{valeur de l'objectif du problème primal pour } x} \leq \underbrace{\sum_{i=1}^{m} b_i u_i}_{\text{valeur de l'objectif du problème dual pour } y}$$

Preuve

On sait que $\sum_{j=1}^{n} a_{ij} x_j \leq b_i \; \forall i \in \{1, \dots, m\}$
On sait que $\sum_{i=1}^{n} a_{ij} u_i \geq c_j \; \forall j \in \{1, \dots, n\}$

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} u_i\right) x_j$$

$$\sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j\right) u_i \leq \sum_{i=1}^{m} b_i u_i$$

Variables

- $y_j = \begin{cases} 1 & \text{si l'entrepôt } j \text{ est construit} \\ 0 & \text{sinon} \end{cases} \quad \forall j \in \{1, ..., m\}$ — Nombre d'entrepôts potentiels
- $x_{ij} = \begin{cases} 1 & \text{si } i \text{ est approvisionné par l'entrepôt } j \\ 0 & \text{sinon} \end{cases} \quad \forall i \in \{1, ..., n\}, \ \forall j \in \{1, ..., m\}$ — Nombre de villes

Modèle mathématique

$$\begin{cases} \text{minimiser} & z = \sum_{j=1}^{m} (f_j y_j + \sum_{i=1}^{n} c_{ij} x_{ij}) \\ \text{tel que} & \sum_{j=1}^{m} x_{ij} = 1 \quad \forall i \\ & x_{ij} \leq y_j \quad \forall i, j \\ & y_j \in \{0, 1\} \quad \forall i \\ & x_{ij} \in \{0, 1\} \quad \forall i, j \end{cases}$$

Relaxation linéaire

2ème idée : Énumération implicite par encadrement de la valeur optimale

Définition - **Relaxation continue** d'un problème entier (P)

Problème obtenu lorsqu'on "oublie" le caractère entier des variables
Ex : $x \in \{1, 2, ..., n\} \rightarrow x \in [1, n]$

Intérêts

- On obtient un PL continu qu'on sait résoudre
  Par exemple par le simplexe
- Fournit une borne sur la valeur optimale

Exemple - Relaxation linéaire du modèle de localisation d'entrepôt

$$\begin{cases} \text{minimiser} & z = \sum_{i=1}^{n} f_i y_i + \sum_{j=1}^{n} c_{ij} x_{ij} \\ \text{tel que} & \sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \\ & x_{ij} \leq y_i \quad \forall i, j \\ & y_i \in [0, 1] \quad \forall i \\ & x_{ij} \in [0, 1] \quad \forall i, j \end{cases}$$

Exemple

$$\begin{array}{lrrrrl} \max z = & 9x_1 & +5x_2 & +6y_1 & +4y_2 & \\ \text{s.c.} & & & y_1 & +y_2 & \leq 1 \\ & & & y_1 & & \leq x_1 \\ & & & & y_2 & \leq x_2 \\ & 6x_1 & +3x_2 & +5y_1 & +2y_2 & \leq 10 \\ & x_1, & x_2, & y_1, & y_2 & \in \{0, 1\} \end{array}$$

Optimum continu

- $(x_1, x_2, y_1, y_2) = (\frac{5}{6}, 1, 0, 1)$
- $Z_c = \frac{33}{2}$

Conclusion

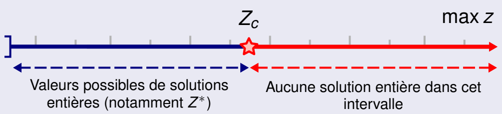
Optimum entier $\leq \frac{33}{2}$

Relaxation continue : interprétation

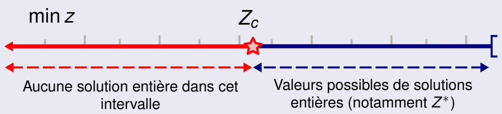
Propriété générale - Relaxation continue

Optimum entier Optimum continu

- Pour un problème de maximisation : $Z^* \leq Z_c$
  Z_c est une borne supérieure de Z^*



- Pour un problème de minimisation : $Z^* \geq Z_c$
  Z_c est une borne inférieure de Z^*



Méthode de résolution de PLNE

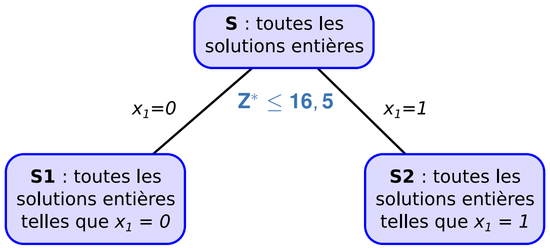
Algorithme de ***branch-and-bound***
Séparation et évaluation en français

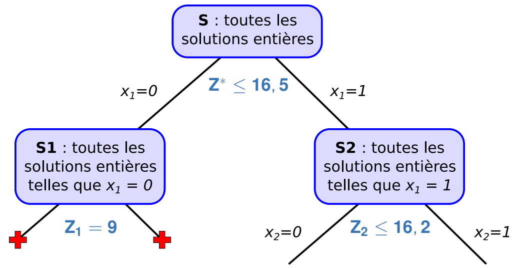
Principe

- Encadrement de la valeur optimale
  Borne inférieure et supérieure
- Énumération limitée afin d'obtenir un encadrement de plus en plus fin

Branch and bound - Exemple

- Solution de la relaxation continue :
  - $(x_1, x_2, y_1, y_2) = (\frac{5}{6}, 1, 0, 1)$
  - $Z_c = 16,5 \geq Z^*$
- x_1 n'est pas entier
- On va « brancher » selon les deux valeurs possibles de x_1 : 0 et 1





Conclusions actuelles

- Valeur de la meilleure solution admissible connue : 9
  Solution courante
- Meilleure borne supérieure connue : 16,2

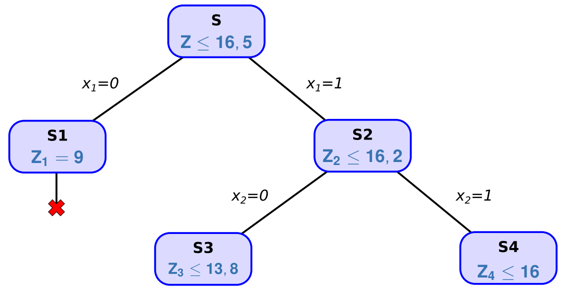
La valeur optimale est donc comprise entre 9 et 16,2

Comment continuer ?

- Élaguer la branche de S1
  Car solution entière trouvée
- Brancher en S2
  Car solution fractionnaire trouvée

Sur quelle variable brancher en S2 ?

- Solution de la relaxation continue : $(x_1, x_2, y_2, y_2) = (1, \frac{4}{5}, 0, \frac{4}{5})$
- On peut brancher sur x_2 ou y_2
  Car valeurs fractionnaires

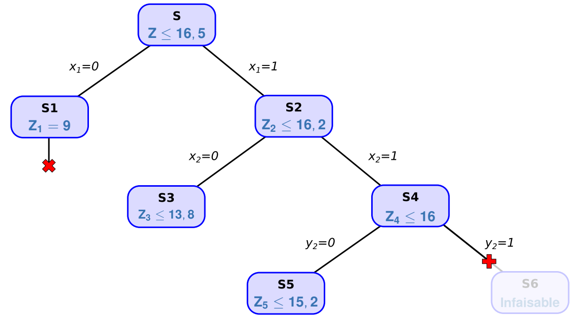


Conclusions actuelles

- Valeur de la meilleure solution admissible connue : 9
- Meilleure borne supérieure connue : 16

Comment continuer ?

- On ne peut élaguer ni S3 ni S4
- On branche en S4 car il possède la plus grande borne supérieure
  Peut potentiellement contenir une solution admissible de valeur $> 13,8$, contrairement à S3
  - On branche sur y_2 qui est fractionnaire en S4



QCM

A ce stade, que peut-on élaguer ?

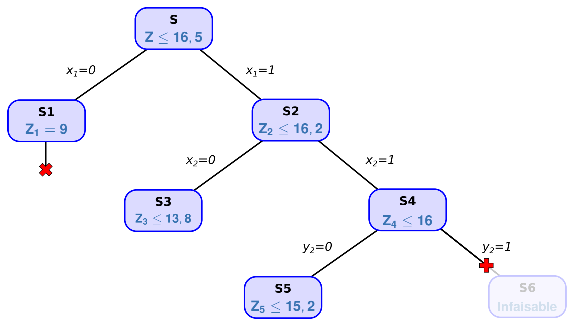
(A) S3 seul

(B) S5 seul

(C) S3 et S5

(D) ni S3 ni S5

ni S3, ni S5, il peuvent encore tous deux contenir l'optimum situé entre 9 et 15.2



Conclusions actuelles

- Valeur de la meilleure solution admissible connue : 9
- Meilleure borne supérieure connue : 15,2

Comment continuer ?

- On branche en S5 qui a la plus grande borne supérieure
  Peut potentiellement contenir une solution admissible de valeur $> 13,8$
  - On branche sur y_2 qui est fractionnaire en S5

Un tout petit peu de combinatoire

Durée d'exécution d'un algorithme

Taille des données d'entrée

Supposons 1 étape = 1μs

Nombre d'étapes élémentaires

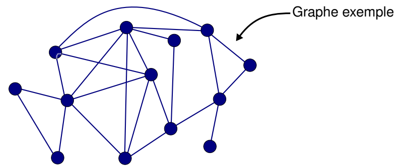
n	log(n)	n	n^2	2^n	n!
10	10^{-6}	10^{-5}	10^{-4}	10^{-3}	4
25	10^{-6}	2.10^{-5}	6.10^{-4}	30	2.10^{19}
50	2.10^{-6}	5.10^{-5}	3.10^{-3}	10^9	3.10^{58}

En secondes

Polynomial ("efficace") — Exponentiel ("non efficace")

Complexité

- Problème "facile"
  Peut se résoudre de façon exacte par un algorithme polynomial
- Problème "difficile"
  Seuls algorithmes connus pour les résoudre de façon exacte sont "exponentiels"

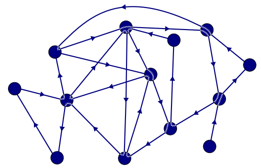


Définition : Chaîne eulérienne

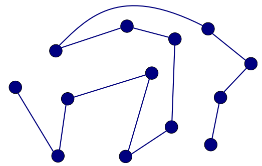
Chaîne passant exactement une fois sur chaque arête

Définition : Chaîne hamiltonienne

Chaîne passant exactement une fois par chaque sommet



Exemple de chaîne eulérienne



Exemple de chaîne hamiltonienne

Trouver une chaîne eulérienne

Problème **"facile"**

Trouver une chaîne hamiltonienne

Problème **"difficile"**

Comment reconnaître la difficulté d'un problème ?

Théorie de la complexité

Attention : ce cours n'en donne qu'une idée **intuitive**

Définition - **Problème de décision**

Problème dont la réponse est oui ou non

Définition intuitive - Problème de décision P de **classe NP**

Si vous savez que P a pour réponse oui, il est facile d'en convaincre quelqu'un d'autre
Mais trouver que la réponse est oui peut rester difficile

Exemple

Si on connaît un cycle hamiltonien, il est facile de convaincre quelqu'un qu'il en existe un
Mais trouver un cycle hamiltonien peut être difficile

Le problème SAT ("satisfiabilité" d'une expression)

Problème de décision SAT

Fonction booléenne

Existe-t-il une affectation des variables telle que f soit vraie ?

Exemple

$f(x) = (x_1 + \overline{x}_2 + x_3)(\overline{x}_1 + \overline{x}_2 + x_4) + (x_2 + \overline{x}_3 + x_4)(x_1 + x_3 + \overline{x}_4)$

- une solution : $(x_1, x_2, x_3, x_4) = ($vrai, faux, vrai, vrai$)$

- Stephen Cook a classé le problème SAT comme NP-complet
- SAT est le premier problème NP-complet connu