ROPL

Chapitre I : Programmation linéaire en 2D

Problème A

> Bob fabrique des yaourt de deux type : Allégés et sucrés, avec 3 ingrédients. Les proportions sont les suivantes :
>
> | Ingrédients/Type | Allégés | Sucrés |
> | --- | --- | --- |
> | Fraises | 2 | 1 |
> | Lait | 1 | 2 |
> | Sucre | 0 | 1 |
>
> Prix de vente :
>
> - Allégés : 4€/kg
> - Sucrés 5€/kg
>
> Les stockes disponibles sont :
>
> - 800kg de fraises
> - 700L de lait
> - 300kg de sucre
>
> **Question :** Comment maximiser le revenu de Bob ?

Modélisation problème A

Soit X_a la quantité produite de yaourt allégé et X_S la quantité produite de sucré

Fonction objectif :

$$max(4x_a + 5x_s)$$

Ce qui va correspondre au revenu de Bob.

Contraintes :

$$\begin{gathered} 2x_a + x_s \leq 800 \text{ (800kg de farine)} \\ x_a + 2x_s \leq 700 \text{ (800kg de lait)} \\ x_s \leq 300 \text{ (300kg de sucre)} \\ x_a \geq 0 \\ x_s \geq 0 \end{gathered}$$

Résolution graphique

Voir cours du prof pour la courbe.

- *Droite $2x_a + x_s = 800$* : deux points (0,800) et (400,0)
    - **Remarque :** 2*0 + 0 = 0 < 800, donc 0 est du côté <= 800
- *Droite $x_a + 2x_s = 700$* : deux points (700,0) et (0,350)

Domaine réalisable : Ensemble des solutions réalisables

Remarque : le maximum s'il existe, est atteint en un sommet du domaine réalisable

Conséquence : Pour trouver le maximum s'il existe, il suffit de calculer la valeur de la fonction objectif pour chaque sommet.

Sommet	O: (0,0)	A:(0,300)	B(100,300)	C : (300, 200)	D : (400, 0)
Valeurs ($4x_a+5x_s$)	0	1500	1900	2200	1600

Donc Bob gagnera au maximum 2200 € en faisant 300 allégés et 200 sucrés.

Il s'agit ici d'un problème de **production**.

Etapes pour résoudre un problème d'optimisation

1. Modélisation
   - Quelles sont les variables a introduire ?
   - Quelle est la fonction objectif ?
   - Quelles sont les contraintes ?
2. Résolution des programmes linéaires (PL) obtenu
   - En 2D : Résolution graphique
   - En général : Algo du simplexe

Un peu de vocabulaire

Un programme linéaire est généralement représenté sous forme matricielle :

$$max(c.x)$$
$$\begin{cases} A_x \leq b \\ x \geq 0 \end{cases}$$

Dans le problème de Bob :

$$x = \begin{pmatrix} x_a \\ x_s \end{pmatrix}$$

c = [4 5], donc c.x = $4x_a + 5x_s$ et

$$a = \begin{pmatrix} 2\ 1 \\ 1\ 2 \\ 0\ 1 \end{pmatrix} et\ b = \begin{pmatrix} 800 \\ 700 \\ 300 \end{pmatrix}$$

$$A_x = \begin{pmatrix} 2\ 1 \\ 1\ 2 \\ 0\ 1 \end{pmatrix} \begin{pmatrix} x_a \\ x_s \end{pmatrix} = \begin{pmatrix} 2x_a + x_s \\ x_a + 2x_s \\ x_s \end{pmatrix} \leq \begin{pmatrix} 800 \\ 700 \\ 300 \end{pmatrix}$$

Définition :

$$\{x : A_x \leq b, x \geq 0\}$$

est l'ensemble des solutions réalisables et appelé **POLYEDRE**.

C'est aussi l'intersection d'un nombre fini de demi-espaces. Un polyèdre borné est un **POLYTOPE**. En 2D, ces demi-espaces sont des demi-plans et les polytopes sont polygones.

Une **FACE** d'un polèdre est l'ensemble des points du polyèdre qui vérifie une des inégalités à égalité.

Si lorsqu'on enlève l'inégalité de la description (A_x <= b), on obtient le même polyèdre, cette inégalité est dite **REDONDANTE**.

Remarque : Les polyèdres sont convexes. P convexe lorsque pour tout x, y dans P, le segemnt [x, y] est contenu dans P.

Complexité de l'algo

Quelle est la complexité de l'algorithme "résolution graphique" (algorithme utilisé un peu plus haut) ? Appliqué a un polygone définit par un m inégalités, en supposant qu'aucune inégalité n'est redondante.

Mini-question : Combien P a-t-il de sommets ? ~> m

Algo :

> Pour toute paire de droite provenant de la description du polygone, on calcule le point d'intersection.
>
> Si ce point est dans ce polygone, c'est un sommet

O(m^2)

> Ensuite, il suffit de trouver un sommet de plus grande valeur

O(m) : Il y a m sommet, et on doit calculer la valeur de la fonction objectif pour chacun d'entre eux.

Total : O(m^3) il est donc polynomial.

<u>Question :</u> Qu'est ce que ça donne en dimension d ?

Si P est un polytope avec m inégalités : m dimension d, un sommet est l'intersection de d de ces m inégalités.

$$\binom{m}{d} = (m^d)$$

Et m^d possibilités ça explose.

Redondance

Sur l'exercice 1 du TD_2D on peut observer que la droite $20x_{T1} + 30x_{T2} \leq 480$ est redondante. Pourquoi ?

On a déjà :

$$\begin{gathered}(1) 40x_{T1} + 30x_{T2} \leq 360 \\ (2) 20x_{T1} + 30x_{T2} \leq 480 \\ (3) - x_{T1} \leq 0 \\ (4) - x_{T2} \leq 0\end{gathered}$$

Sur le dessin on observe que (2) est redondante :

$$\begin{gathered}(1) - (3) : \begin{cases} 40x_{T1} + 30x_{T2} \leq 360 \\ -x_{T1} \leq 0 \end{cases} \\ 39x_{T1} + 30x_{T2} \leq 360 \\ \text{Cette in}_{\acute{e}}\text{galit}_{\acute{e}}\text{ est valide pour l'ensemble des points satifaisant (1) et (3)}\end{gathered}$$

$(1) - 20 * (3)$ donne donc $20x_{T1} + 30x_{T2} \leq 360(*) < 480$.

Y'a t-il une relation entre (*) et (2) ?

Tous les points satisfaisant (*) vérifiant (2).

Idée : Une inégalité est redondante si on peut écrire une inégalité au moins aussi forte en combinant les autres.

Chapitre II : Algorithme du simplexe

Programme linéaire en général

Forme générale

$$min/max \sum_{i=1}^{d} c_i x_i \quad \text{(Avec "d" pour "dimension")}$$
$$\begin{cases} \sum_{j=1}^{d} a_{i,j}x_j \leq b_i, i \in I^{\leq} \\ \sum_{j=1}^{d} a_{i,j}x_j \geq b_i, i \in I^{\geq} \\ \sum a_{i,j}x_j = b_i, i \in I^{=} \\ x_i \geq 0, j \in J_+ \\ x_i \leq 0, j \in J_- \\ x_i \; libre \; 0, j \in J \end{cases}$$
$$\text{(Avec } I^{\leq,\geq,=} \text{ ensemble des indices d'inégalités de type } \leq, \geq ou \; =)$$

Forme canonique

$$max \; c.x$$
$$\begin{cases} A_x \leq b \\ x \geq 0 \end{cases}$$

- A est une matrice de taille $m * d$.
- b est justifié par $\sum_{j=1}^{d} a_{i,j}x_j \leq b_i, i = 1...m$
- 0 est justifié par $x_j \geq 0, j = 1, \ldots d$

Forme standard

$$max \; cx$$
$$\begin{cases} A_x = b \\ x \geq 0 \end{cases}$$

On remarque que le signe (=) de la première égalité est la différence avec la forme canonique.

Conséquence : Sous forme standard on peut supposer $rang(A) = m$, où m est le nombre de lignes de A.

Ex :

> $$\begin{pmatrix} 1 \; 0 \\ 0 \; 1 \\ 1 \; 1 \end{pmatrix} \text{N'a pas de rang(A) = m, car } L_3 = L_1 + L_2$$

Dorénavant on supposera que dans la forme standard le rang de la matrice est égal à son nombre de ligne. $rang(A) = m$.

Théorèmes

- Au prix d'un éventuel ajout de contraintes et de variables, tout programme linéaire peut être transformé en un programme linéaire *équivalent* (toute solution optimale pour l'un fournit la solution optimale pour l'autre) sous forme canonique.
- Pareil pour la forme standard.

Exemple

$$max(x_1 + x_2)$$
$$\begin{cases} x_1 - 2x_2 \leq 1 \\ 2x_1 - x_2 \geq 2 \\ -2x_1 - x_2 \leq 1 \\ x_1, x_2 \geq 0 \end{cases} - \text{Forme standard} \rightarrow \begin{cases} x_1 - 2x_2 + \underline{e_1} = 1 \\ 2x_1 - x_2 - \underline{e_2} = 2 \\ -2x_1 - x_2 + \underline{e_3} = 1 \\ x_1, x_2, \underline{e_1}, \underline{e_2}, \underline{e_3} \geq 0 \end{cases}$$

La base x_B est $\{e_1, -e_2, e_3\}$ avec une solution associée à $(0, 0, 1, -2, 1)$ qui n'est **pas réalisable**.

Règles de transformations

Objet a transformer	Règles de transformations
$min \leftrightarrow max$	$min\ cx = -max(-cx)$
$\geq \leftrightarrow \leq$	$ax \geq b \Leftrightarrow -ax \leq -b$
$= \leftrightarrow \leq$	$ax = b \Leftrightarrow \begin{cases} ax \leq b \\ -ax \leq -b \end{cases} \Leftrightarrow \begin{cases} ax \leq b \\ ax \geq b \end{cases}$
$x_j \leq 0$	On pose $x_j = -x_j$ et on remplace x_j par $-x'_j$ partout, et du coup on a $x'j \geq 0$
$x_j$libre (pas de signe précisé)	On remplace x_j par $x_j^+ - x_j^-$ et on ajoute $x_j^+ \geq 0$ et $x_j^- \geq 0$ (on supprime x_j ensuite) **Exemple :** $x_j = 5 \rightarrow x_j^+ = 5, x_j^- = 0$ $x_j = -3 \rightarrow x_j^+ = 0, x_j^- = 3$
$\leq \longrightarrow =$	Ajout de variable d'écarts $ax \leq b$ devient $\begin{cases} ax + e = b \\ e \geq 0 \end{cases}$ avec e en variable d'écart

Exemple

Forme générale

$$min(x - y)$$
$$\begin{cases} x \geq 1 \\ x + y = 2 \\ x - 2y \leq 4 \\ x \geq 0 \text{ (y est donc libre)} \end{cases}$$

PL équivalent sous forme canonique

$$-max - x + y^+ - y^-$$

$$\begin{cases} -x \leq -1 \\ x + y^+ - y^- \leq 2 \\ -x - y^+ + y^- \leq -2 \\ x - 2y^+ + 2y^- \leq 4 \\ x, y^+, y^- \geq 0 \text{ (y n'est donc plus libre)} \end{cases}$$

PL équivalent sous forme standard

$$-max - x + y^+ - y^-$$

$$\begin{cases} -x + e_1 = -1 \\ x + y^+ - y^- = 2 \\ x - 2y^+ + 2y^- + e_2 = 4 \\ x, y^+, y^-, e_1, e_2 \geq 0 \end{cases}$$

Définitions

Soit :

$$max\ cx$$
$$\begin{cases} A_x = b \\ x \geq 0 \end{cases}$$

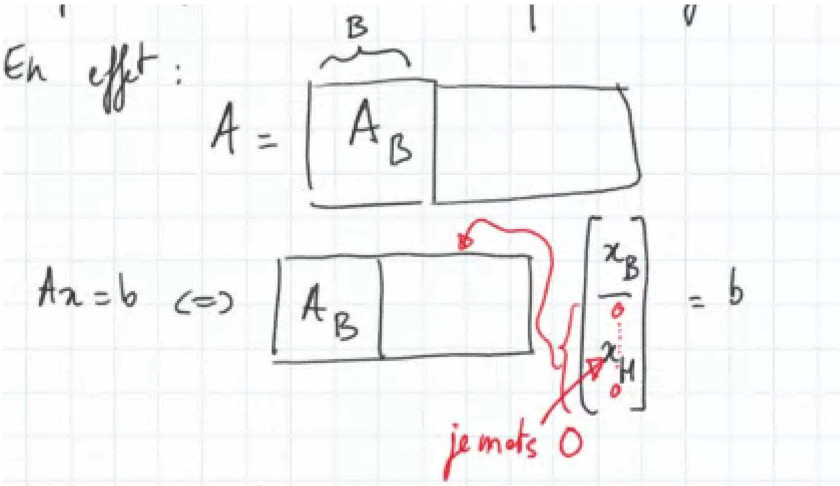
un programme linéaire sous forme standard.

- Un ensemble $B \subseteq \{1, -d\}$ (avec d correspondant au nombre de colonne de la matrice) tel que les colonnes de A indicées par B forment une matrice A_b inversible est appelé **une base**
  - $x_B = (x_j : j \in B)$ sont les **variables de base**
  - $x_H = (x_j : j \notin B)$ sont les **variables hors base**

Ex :

> $$A = \begin{pmatrix} 1\,2\,4\,7 \\ -1\,1\,1\,8 \end{pmatrix}$$
>
> Q: $\{1, 2\}$ forme une base ?
>
> R: Oui car $det \neq 0$
>
> Par rapport à la base {1,2}:
>
> - x_1, x_2: sont les variables de base
> - x_3, x_4 sont hors-base
>
> Q: {2,3} forme une base ?
>
> R: Non car $det = 0$

- Etant donné une base B : poser $x_H = 0$ (c'est à dire mettre toutes les variables hors-base à zéro) définit une solution unique au système $A_x = b$



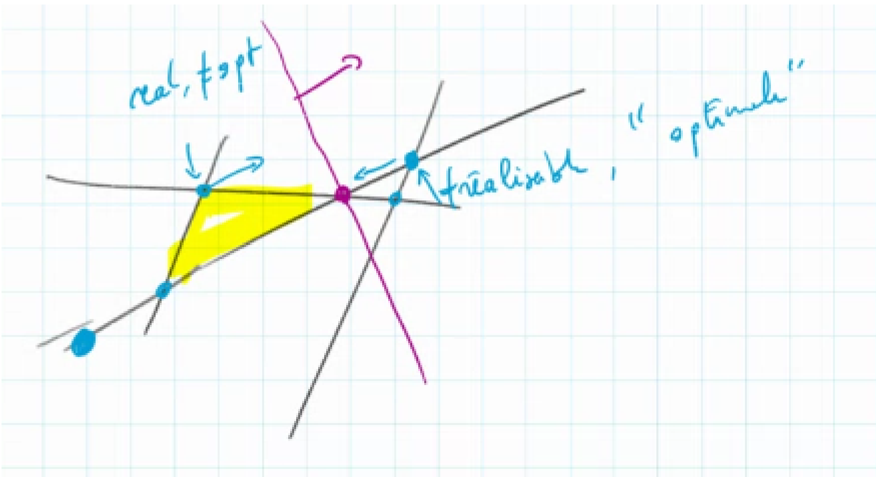
On obtient $A_x x_B = b$ qui a pour unique solution $x_B = A_B^{-1} b$.

Cette solution (x_B, x_H) est appelée **solution de base associée à la base.** (Variable hors base a 0 et système linéaire restant résolu)

Si $x_B, x_H \geq 0$: c'est une **solution de base réalisable**. (Toutes les variables sont positives ou nulles)

Et le simplexe va chercher a améliorer les valeurs de la fonction objectif.

- Coûts réduits : On écrit la fonction objectif en fonction des variables hors base et une fois ceci fait les coefficients obtenus sont les coûts réduits des variables.
    - <u>Idée :</u> Le coût réduit d'une variable indique de combien augmenterais la fonction objectif si en faisant entrer la variable dans la base.
    - <u>Conséquence :</u> Si tout les coûts réduits sont négatifs ou nul alors la solution courante est "optimale" <u>ssi</u> elle est **réalisable**. Si elle n'est pas réalisable cela signifie que la solution courante est du côté de l'optimale mais est en dehors du domaine réalisable.



Et le point rose est **réalisable** et **optimale**.

Déterminer une base et si elle est réalisable

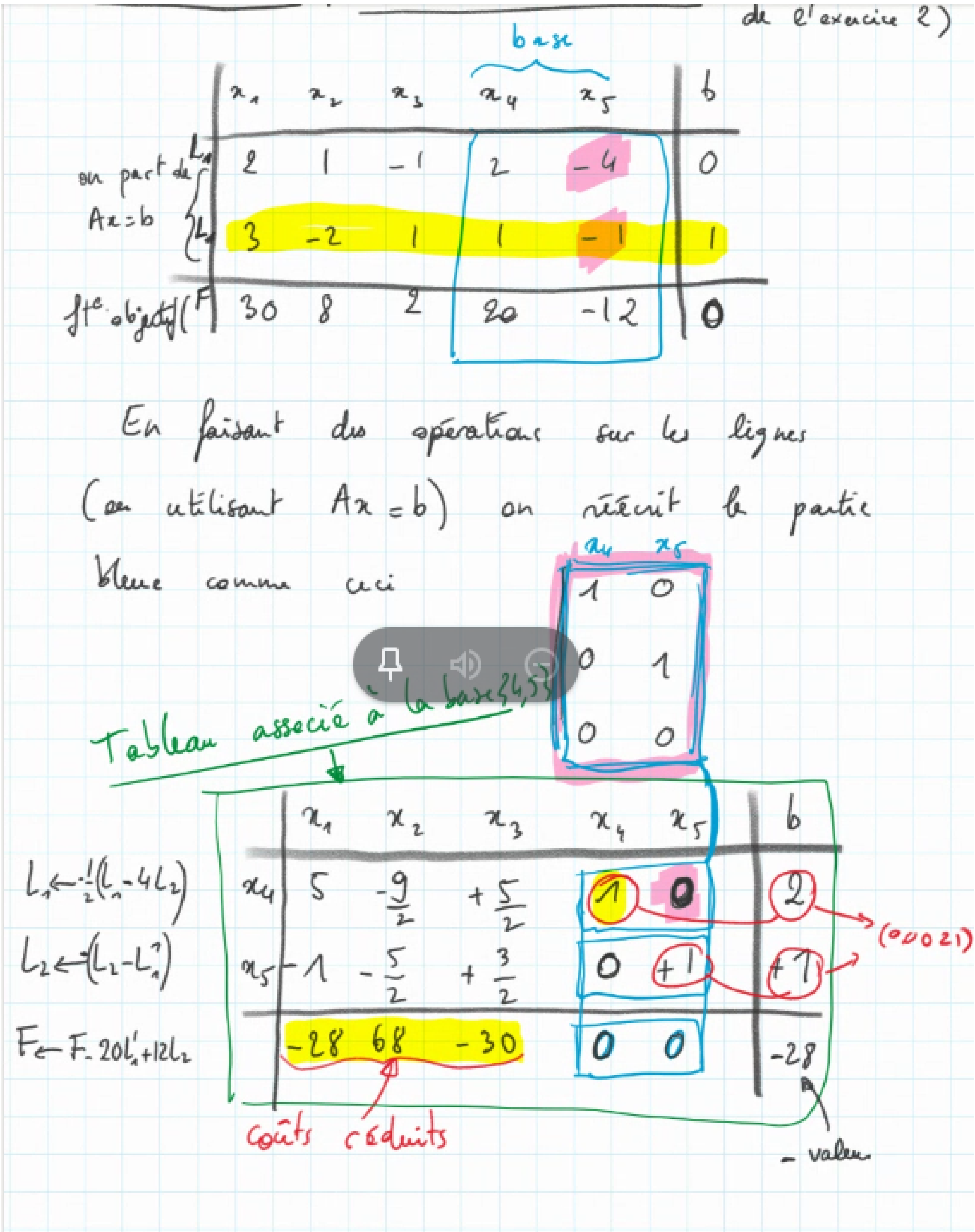
Etant donné un PL, pour vérifier si la base est réalisable :

1. Déterminer la matrice carrée obtenue
2. Si c'est < 0 ça forme une base
3. Déterminer les variables en base (celle dans la matrice carrée) et hors base (les autres)
4. Poser les variables hors bases a 0
5. Résoudre le système linéaire sachant ça
6. Les valeurs trouvées pour les variables en base détermine la solution de base

7. Si elle est supérieur ou égal a 0 alors c'est réalisable

Trouver les coûts réduits

Ici sur la base $I = \{4, 5\}$ issue de l'exercice 2 du TD "Solution de base"



Remarque : On peut lire immédiatement dans le tableau :

- Les coûts réduits
- Les coordonnées de la solution de base associée

Le tableau associé à une base B donnée est la réécriture sous forme de tableau de $A_x = b$ avec F

$\begin{pmatrix} 1\,0 \\ 0\,1 \\ 0\,0 \end{pmatrix}$ dans les colonnes correspondant à B.

Algorithme du simplexe (Phase 2)

Entrée/Sortie

Entrée : Un programme linéaire sous forme standard et une solution de base réalisable (base B) et son tableau

Sortie : La valeur du programme linéaire, et une solution optimale si cette valeur est finie.

Corps

Tant qu'il existe une variable hors-base de coût réduit strictement positif :

- Variable entrante : variable k hors-base de coût réduit maximum
- Variable sortante : variable l minimisant (avec le plus petit) $\frac{b_i}{a_{i,k}} avec\ a_{i,k} > 0$
- Nouvelle base : $B := B \cup \{k\} \setminus \{l\}$
- Ecrire le tableau associé à la nouvelle base

Terminaison (Conditions de sortie)

- Tous les coûts réduits sont négatifs ou nuls : l'algorithme termine et la solution de base courante est optimale
- S'il existe un coût réduit strictement positif (donc il y a une variable entrante) mais que tous les $a_{i,k}$ sont positif ou nul (il n' a pas de variable sortante), alors la valeur du PL est $+\infty$

Remarque : Pour revenir a une solution optimale du PL de départ, il suffit de ne plus tenir compte des variables d'écarts (remarque issue de l'exemple déroulé du simplexe)

Exemple du tableau type

	Nom de toutes les variables	Résultat des équations (b)
Nom des variables de base	Coefficients de toutes les variables	Résultat des équations
Coût (c)	Coefficient de la fonction objectif	Inverse de la solution courante

Exemple tiré de l'exercice 2 du TD :

$$\max \quad x_1 + x_2$$
$$\begin{cases} 2x_1 + 3x_2 \leq 1 \\ x_1 \leq \frac{1}{3} \\ x_2 \leq \frac{1}{4} \\ x_1 \ , \ x_2 \geq 0 \end{cases}$$

Le premier tableau sera donc :

$x_B = \{e_1, e_2, e_3\}$

$x_H = \{x_1, x_2\}$

	x_1	x_2	e_1	e_2	e_3	**b**
e_1	2	3	1	0	0	1
e_2	1	0	0	1	0	$\frac{1}{3}$
e_3	0	1	0	0	1	$\frac{1}{4}$
C	1	1	0	0	0	0

Algorithme du Simplexe (Phase 1)

Entrée/Sortie

Entrée : Un PL sous forme standard

Sortie : Une solution de base réalisable s'il en existe une (sinon le PL est vide et on ne peut pas aller plus loin.)

Corps

- On ajoute une variable artificielle y_i par contrainte $a_i x = b_i$ avec un coefficient devant y_i dépendant de b_i:
  - $+y_i$ si $b_i \geq 0$
  - $-y_i$ si $b_i < 0$

  et $(y_i \geq 0)$
- On résout ce nouveau PL en appliquant la phase II avec comme fonction objectif : $min(y_1 + \ldots + y_m) = -max(-y_1 - \ldots - y_m)$. et comme solution de base réalisable celle associée à la base des variables artificielles $\{y_1, \ldots y_m\}$.

Terminaison (conditions de sortie)

- Si la valeur de de ce nouveau PL est 0, on obtient une solution de base réalisable des PL de départ
- Sinon, le PL de départ est vide

Exemple

$$\begin{gathered} max(x_1 + x_2) \\ \begin{cases} -x_1 + 3x_2 = 2 \\ -7x_1 + 4x_2 = -2 \\ x_1, x_2 \geq 0 \end{cases} \end{gathered}$$

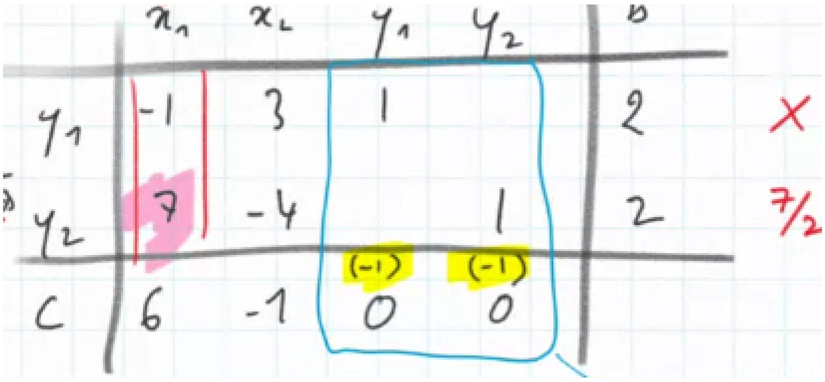
Ce PL est déjà en format std mais il n'y a pas de base réalisable évidente (pas d'identité matricielle évidente donc relou a déterminer) il est donc nécessaire de de passer par la phase 1 du simplexe :

- Introduction des variables artificielles (y_1, y_2)

$$\begin{gathered} -max(-y_1 - y_2) == min(y_1 + y_2) \\ (N) \begin{cases} -x_1 + 3x_2 + \underline{y_1} = 2 \\ -7x_1 + 4x_2 - \underline{y_2} = -2 \\ x_1, x_2, \underline{y_1}, \underline{y_2} \geq 0 \end{cases} \end{gathered}$$

Base $\{y_1, y_2\}$, solution associée (0,0,2,2) réalisable.

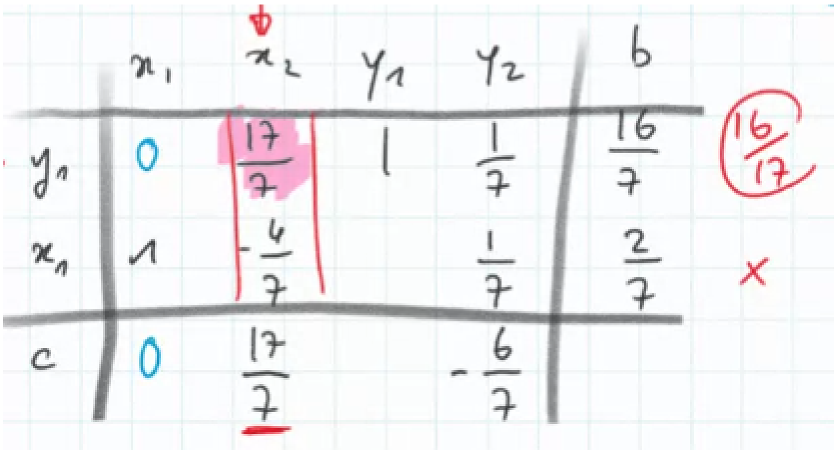
- On résout (N) en appliquant la **phase II**
  - Tableau associé à $\{y_1, y_2\}$



	x_1	x_2	y_1	y_2	b	
y_1	-1	3	1		2	X
y_2	7	-4		1	2	7/2
			(-1)	(-1)		
c	6	-1	0	0		

La variable qui sort est donc y_2 au profit de x_1. ($\frac{7}{2} > NaN$)

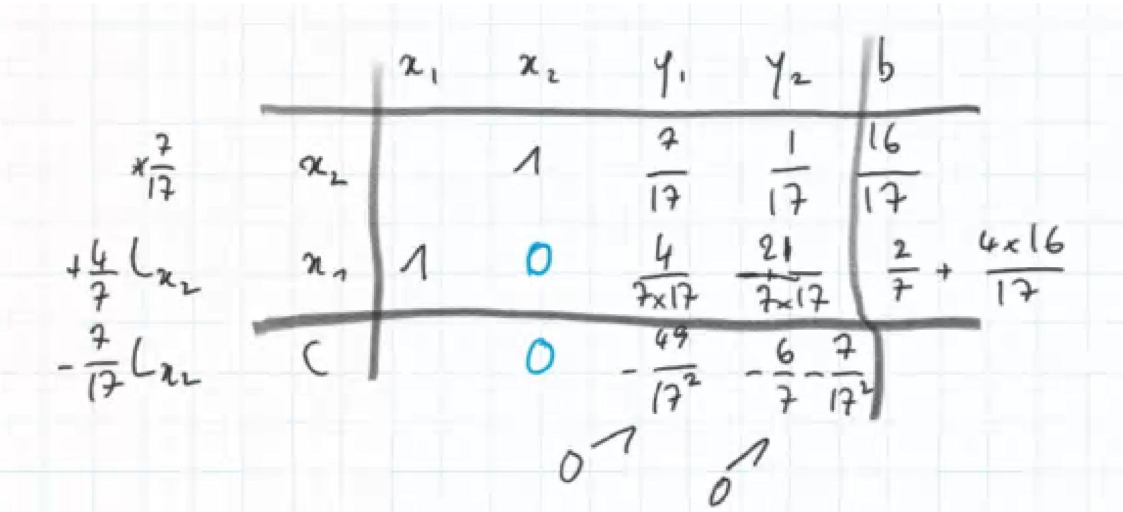
  - Tableau associé à $\{y_1, x_1\}$



	x_1	x_2	y_1	y_2	b	
y_1	0	$\frac{17}{7}$	1	$\frac{1}{7}$	$\frac{16}{7}$	$\frac{16}{17}$
x_1	1	$-\frac{4}{7}$		$\frac{1}{7}$	$\frac{2}{7}$	X
c	0	$\frac{17}{7}$		$-\frac{6}{7}$		

La variable qui sort est donc y_1 au profit de x_2. ($\frac{16}{17} > NaN$)

  - Tableau associé à $\{x_2, y_2\}$

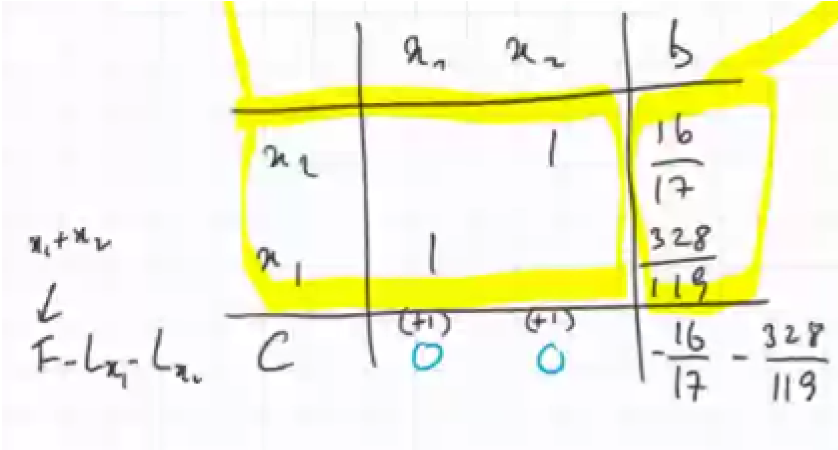


		x_1	x_2	y_1	y_2	b
$\times\frac{7}{17}$	x_2		1	$\frac{7}{17}$	$\frac{1}{17}$	$\frac{16}{17}$
$+\frac{4}{7}L_{x_2}$	x_1	1	0	$\frac{4}{7\times 17}$	$\frac{21}{7\times 17}$	$\frac{2}{7} + \frac{4\times 16}{17}$
$-\frac{7}{17}L_{x_2}$	C		0	$-\frac{49}{17^2}$	$-\frac{6}{7}-\frac{7}{17^2}$	
				$0 \nearrow$	$0 \nearrow$	

Tous les coûts réduits sont négatifs ou nuls : la solution optimale de la phase I est ($\frac{328}{119}, \frac{16}{17}, 0, 0$) de valeur 0.

Autrement dit on a une base x_1, x_2 du PL de départ avec solution de base associée : ($\frac{328}{119}, \frac{16}{17}$) réalisable est la variables artificielles sont nulles.

- On donc lancer les phase II pour le PL de départ
  - Tableau associé à $\{x_1, x_2\}$:



La partie $A_x = b$ (Qui pour rappel correspond ici a : $\{x_1, x_2\}$) de tableau est déjà écrite à la dernière étape de la phase I.

Bilan

Etant donné un PL en général pour le résoudre :

1. On le met sous forme standard
2. S'il n'y a pas de solution de base **réalisable** et **<u>évidente</u>** on applique la phase I, s'il y'en a une on saute directement a l'étape (3)
   - Soit le PL de départ obtenu est vide dans ce cas on **s'arrête** (Optimum de la phase I différent de 0)
   - Soit on obtient une solution de base B réalisable du PL de départ
3. On applique la phase II du PL de départ avec B comme base réalisable.
   - Soit la valeur est $+\infty$ (Lorsqu'il y'a une variable entrante mais pas de variable sortante)
   - Soit la on obtient une solution réalisable optimale et sa valeur (Lorsqu'il y'a pas de variable entrante)

Variable d'écarts VS variables artificielles

$$\begin{gathered} max(x_1 + x_2) \\ \begin{cases} x_1 - 2x_2 \leq 1 \\ 2x_1 - x_2 \geq 2 \\ -2x_1 - x_2 \leq 1 \\ x_1, x_2 \geq 0 \end{cases} - \text{Forme standard} \rightarrow \begin{cases} x_1 - 2x_2 + \underline{e_1} = 1 \\ 2x_1 - x_2 - \underline{e_2} = 2 \\ -2x_1 - x_2 + \underline{e_3} = 1 \\ x_1, x_2, \underline{e_1}, \underline{e_2}, \underline{e_3} \geq 0 \end{cases} \end{gathered}$$

La base x_B est $\{e_1, -e_2, e_3\}$ avec une solution associée à $(0, 0, 1, -2, 1)$ qui n'est **pas réalisable**.

On va donc utiliser la phase I, l'objectif n'est pas d'ajouter des variables artificielles pour rien, mais seulement aux endroits où on en a besoin.

$$\begin{gathered} -max(-y) \\ \begin{cases} x_1 - 2x_2 + e_1 = 1 \\ 2x_1 - x_2 - e_2 + \underline{y} = 2 \\ -2x_1 - x_2 + e_3 = 1 \\ x_1, x_2, e_1, e_2, e_3, \underline{y} \geq 0 \end{cases} \end{gathered}$$

On ne met donc une variable artificielle seulement au niveau de e_2

Avec comme base réalisable de départ $\{e_1, y, e_3\}$ (solution associée $(0, 0, 1, 0, 1, 2)$ réalisable)

Chapitre III : Flots

Définitions

Réseau de transport

C'est un graphe orienté dans lequel il existe :

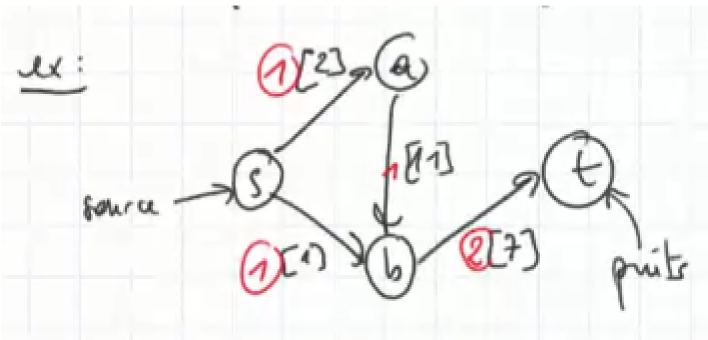
- Un sommet sans prédécesseur, la source
- Un sommet sans successeurs : le puit

et chaque arc est muni d'une capacité de $c_a \geq 0$.

Flot

Un flot est la donnée d'une valeur f_a sur chaque arc a telle que :

1. $f_a \geq 0$, pour tout arc $a \in A$. **Positivité**.
2. $f_a \leq c_a$ pour tout arc $a \in A$. **Respect des capacités**.
3. $\sum_{a \in f^-(n)} f_a = \sum_{a \in f^+(n)} f_a$, pour tout sommet $u \neq s, t$ **Conservation du flot, loi des noeuds, loi de Kirchhoff**



Les valeurs en rouge forment un flot:

- Positif
- Respecte les capacités
- Conservation du flot

de valeur 2.

La **valeur d'un flot** est la quantité de flot sortant de la source.

Conséquence : Le flot sortant de la source est égal au flot entrant dans le puits.

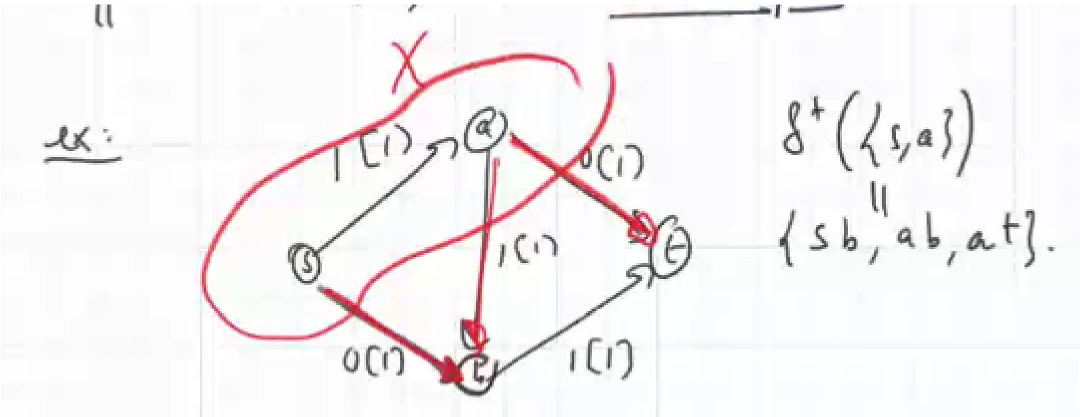
Problématique du chapitre : Déterminer un flot de valeurs maximum

C'est qu'on appellera un **flot maximum**.

Comment trouver les bonnes bornes supérieur sur la valeur d'un flot maximum ?

Remarque : La valeur d'un flot est toujours inférieur ou égale à la somme des capacités des arcs sortant de la source.

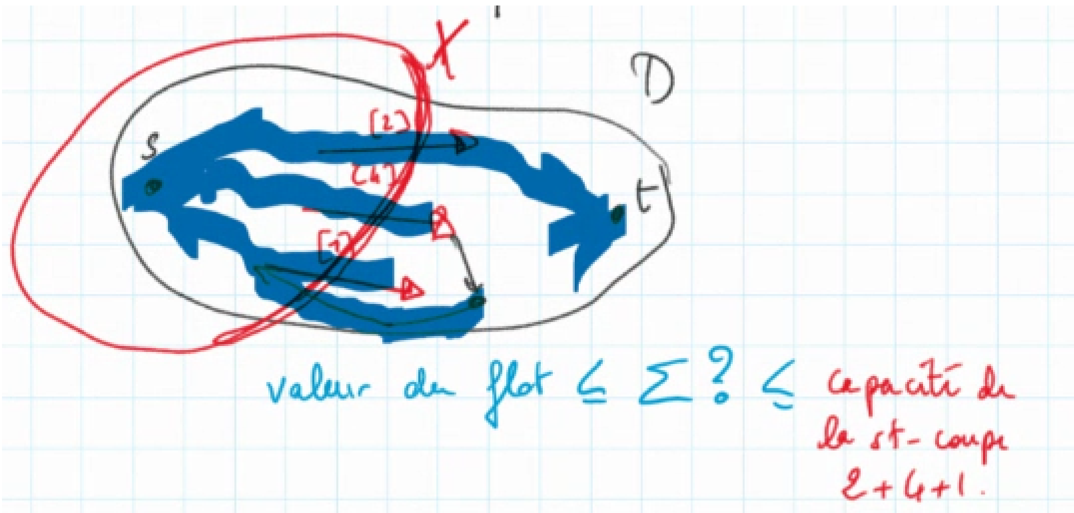
Définition : Soit X un ensemble de sommets contenant la source mais pas le puits on appelle $\delta^+(X)$ une **st-coupe**.



Remarque : Que se passe t-il si on supprime tous les arcs d'une st donnée ?

Après coup, il n'y a plus de chemin de s à t. En effet toute st-coupe intersecte tout chemin de s à t.

Théorème : La valeur d'un flot quelconque est inférieur ou égale à la capacité de toute st-coupe.



Conséquence : Si la valeur d'un flot f est égale à la capacité d'une st-coupe $\delta^+(X)$, alors f est un flot maximum.

Flot max coupe minimum

Chaine augmentante

On se donne un flot f dans un réseau de transport D

A récupérer sur le cours du prof

Algorithme

Entrée/Sortie

Entrée : Un réseau de transport $D_{s,t\in V} = (V, A)$ éventuellement parcouru d'un flot $c_a \longrightarrow \mathbb{R}_+$

Sortie : Un flot de valeur maximum et une st-coupe capacité minimum (égale à la valeur d'un flot)

Corps

<u>MARQUAGE (routine) :</u>

Initialisation : Marquer S du label +.

Itération :

Tant qu'il existe un sommet u marqué non traité :

Traiter u, c'est à dire :

- Pour tout successeur v et u non marqué avec (v,u) non saturé : Marquer v du label +u
- Pour tout prédécesseur w et u non marqué avec (w,u) transportant du flot :
  - Marquer w du label -u

Fin tant que.

Si le puits est marqué, alors on a trouvé une chaine augmentante : il suffit de remonter les labels en partant du puits.

Si le puits n'est pas marqué, alors le flot est maximum, et une st-coupe $\delta^+(X)$ de capacité minimum est donnée par l'ensemble X des sommets marqués.

Ford-Fulkerson

Tant que le puits est marqué lors de MARQUAGE:

- Augmente le flot à l'aide de la chaine augmentante correspondante

Fin tant que.