Programmation linéaire (suite)

1. Le tableau du simplexe (version perso)

Pour résoudre de plus grands problèmes linéaires il faut renoncer au traitement à la main.

Exemple 1.

On considère le programme linéaire suivant

Maximiser z=-$x_1 + 2x_2 - 4x_3$+5

sous les contraintes

$$\begin{cases} x_1 + 3x_2 + 2x_3 + x4 = 9 \\ x_1 + 2x_2 - x_3 + x5 = 2 \\ -x_1 + x_2 + x_3 + x6 = 4 \\ x_1, x_2, x_3, x4, x5 \geqslant 0 \end{cases}$$ qui se paramétrisent comme suit

$$\begin{cases} x4 = 9 - x_1 - 3x_2 - 2x_3 \\ x5 = 2 - x_1 - 2x_2 + x_3 \\ x6 = 4 + x_1 - x_2 - x_3 \\ x_1, x_2, x_3, x4, x5 \geqslant 0 \end{cases}.$$

	$x1$	$x2$	$x3$	$x4$	$x5$	$x6$	z	Memdroite
$x4$	1	3	2	1	0	0	0	9
$x5$	1	2	-1	0	1	0	0	2
$x6$	-1	$+1$	$+1$	0	0	1	0	4
z	1	-2	4	0	0	0	1	5

Les 3 premières lignes se lisent comme les équations

les « 1 » disent les variables de la base : x4,x5,x6

et on retrouve les variables de base dans la colonne de gauche

la 4 ème se lit z+x1-2x2+4x3=5; OK !

1. On regarde les coeff strictement négatifs dans la quatrième ligne, c'est à dire dans z; ce sont eux qui peuvent faire grandir z

S'ils sont tous positifs (ou nuls), c'est fini z est maximal; s'il y a des strictement négatifs, prenons le plus grand en valeur absolue (ici un seul: 2), cela nous définit la colonne de x2, x2 va entrer dans la base .

2. Cherchons dans la colonne de x2 qui va sortir

on cherche $\min\{b_i/a_{i2}, a_{i2} > 0\}$, ce qui détermine une ligne : i_0

Ce sera la case du pivot $(i_0, 2)$ ici: ligne 2 qui correspond à x5, c'est x5 qui va sortir de la base

3. On divise toute la ligne par a_{i2}

d'où le tableau

	$x1$	$x2$	$x3$	$x4$	$x5$	$x6$	b
$x4$	1	3	2	1	0	0	9
$x5$	$1/2$	1	$-1/2$	0	$1/2$	0	1
$x6$	-1	$+1$	$+1$	0	0	1	4
	-1	2	-4	0	0	0	5

puis, comme on veut que x2 soit dans la base, il faut annuler les autres coeff de cette colonne:

4. Pour avoir un « 0 » à la première ligne

L1:=L1-3L2

ce qui donne

	$x1$	$x2$	$x3$	$x4$	$x5$	$x6$	b
$x4$	$-1/2$	0	$7/2$	1	$-3/2$	0	6
$x5$	$1/2$	1	$-1/2$	0	$1/2$	0	1
$x6$	-1	$+1$	$+1$	0	0	1	4
	-1	2	-4	0	0	0	5

5. Pour avoir un « 0 » à la troisième ligne

L3:=L3-L2

ce qui donne

	$x1$	$x2$	$x3$	$x4$	$x5$	$x6$	b
$x4$	$-1/2$	0	$7/2$	1	0	0	6
$x5$	$1/2$	1	$-1/2$	0	$1/2$	0	1
$x6$	$-3/2$	0	$3/2$	0	$-1/2$	1	3
	-1	2	-4	0	0	0	5

6.

Pour avoir un « 0 » à la quatrième ligne

L4:=L4-2L2

ce qui donne

	$x1$	$x2$	$x3$	$x4$	$x5$	$x6$	b
$x4$	$-1/2$	0	$7/2$	1	0	0	6
$x5$	$1/2$	1	$-1/2$	0	$1/2$	0	1
$x6$	$-3/2$	0	$3/2$	0	$-1/2$	1	3
	-2	0	-2	0	-1	0	3

7. On corrige la première colonne pour avoir la liste actualisée des variables de la base

	$x1$	$x2$	$x3$	$x4$	$x5$	$x6$	b
$x4$	$-1/2$	0	$7/2$	1	0	0	6
$x2$	$1/2$	1	$-1/2$	0	$1/2$	0	1
$x6$	$-3/2$	0	$3/2$	0	$-1/2$	1	3
	0	0	3	0	1	0	7

Il n' y a plus de coeff strictement négatifs dans la ligne du gain

z=-3x2-x5+7

Définition 2. *Le tableau du simplexe*

Soit le programme linéaire suivant

$$\text{Maximiser } {}^t\text{cX sous les contraintes } \left\{ \begin{array}{l} \text{AX} + Y = b \\ \begin{pmatrix} X \\ Y \end{pmatrix} \geqslant 0 \end{array} \right., \text{ où } \begin{pmatrix} X \\ Y \end{pmatrix} \geqslant 0 \text{ signifie } \left\{ \begin{array}{l} x_1 \geqslant 0 \\ \ldots \\ \ldots \\ x_n \geqslant 0 \\ y1 \geqslant 0 \\ \ldots \\ \text{ym} \geqslant 0 \end{array} \right. .$$

Le tableau sera le suivant

m+1 lignes, n+m+1 colonnes

(ligne des variables)	$(x1)$	$(x2)$	.	.	.	.	(xn)	$(y1)$	$(y2)$	.	.	(ym)	z	(memdroite)
(ligne $L1$)	$a11$	$a12$	...	...	.	.	$a1n$	1	0	...	.	0	0	$b1$
...	$a21$	$a22$	...	...	...	...	$a2n$	0	1	...	...	...	0	$b2$
...	....	...	...	....	....	...	....	0	0	...	.	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	0	...	...
(ligne Lm)	am1	am2	...	.	...	.	amn	.	...	.	0	1	...	bm
(ligne de z)	$-c1$	$-c2$	...	...	...	...	$-$cn	0	0	...	0	0	1	z(au pt de base)

Si on regarde bien les lignes L1 jusqu'à Lm reproduisent la contrainte AX+Y=b

la ligne des inconnues décrit $\begin{pmatrix} X \\ Y \end{pmatrix}$ et précise qui est multiplié par les coeff de chaque colonne

la ligne de z donne l'expression de z en fonction des inconnues (x1,..,xn,y1,.,ym)

(au début seulement les xi, ensuite au gré des pivotages) .

Théorème 3. *La méthode du simplexe avec le tableau du simplexe*

Nous désignerons les termes des lignes 1 à m, colonnes 1 à n+m par m_{ij} *et la dernière colonne par P:*

1. Si les coefficients de la ligne de z sont $\geqslant 0$ *la valeur de z au point de base est le maximum*

2. Si il existe un coefficient de z qui est <0 *on regarde sa colonne, qui sera désignée dans ce qui suit par* $\begin{pmatrix} m_{1j} \\ \cdots \\ . \\ m_{\mathrm{mj}} \end{pmatrix}$

2.1 si pour tout i $m_{\mathrm{ij}} \leqslant 0$ *z n'est pas borné*

2.2 si il existe $m_{ij} > 0$, *soit* i_0 *tel que* $\frac{p_{i_0}}{m_{i_0 j}} = \min \left\{ \frac{p_i}{m_{ij}} \right\}$, *pivoter autour du terme* $m_{i_0 j}$

c'est à dire :

i) $L_{i_0} \longrightarrow \frac{1}{m_{i_0 j}} L_{i_0}$ *(sur toute la longueur , c'est à dire de* $m_{i_0 1}$ *à* p_{i_0}*)*

ii) pour toute les lignes (de L1 à Ligne de z incluse, sauf la ligne Li0) $L_i \longrightarrow L_i - \frac{m_{\mathrm{ij}}}{} L_{i_0}$

(ainsi dans la colonne j se retrouve un « 1 » et le reste des valeurs numériques est nul.

Ceci a permis l'entrée de la variable de la colonne j dans la base et la sortie de la variable de la ligne i_0 *de cette base.*

et on continue...

Problème 1.

Résoudre, en utilisant le tableau du simplexe, le programme lineaire suivant

Maximiser $f: (x1, x2) \longmapsto 2x1 + x2$

sous les contraintes $\begin{cases} x1\text{-x2} \leqslant 3 \\ x1 + 2x2 \leqslant 6 \\ \text{-x1} + 2x2 \leqslant 2 \\ x1 \geqslant 0, x2 \geqslant 0 \end{cases}$

2. Difficultés possibles pendant l'exécution de la méthode

2.1 *D* n'est pas borné et la restriction de z à D n'est pas bornée

Considérons le programme linéaire suivant:

$$\text{Maximiser } x_1 \text{ sous les contraintes } \begin{cases} x_1 - x_2 + x_3 = 1 \\ -x_1 + x_2 + x_4 = 2 \\ x_1 \geqslant 0, \ldots, x_4 \geqslant 0 \end{cases}.$$

a.

$z = x_1$

$$\begin{array}{l} x_3 = 1 - x_1 + x_2 \\ x_4 = 2 + x_1 - x_2 \end{array}$$

b.

$z = 1 + x_2 - x_3$

$$\begin{array}{l} x_1 = 1 + x_2 - x_3 \\ x_4 = 3 - x_3 \end{array}$$

. Troisième :

La seule variable qui puisse entrer dans la base est x_2, qui ne sera soumis à aucune majoration (cf les expressions de x_1, x_4) et donc x_2 peut prendre des valeurs aussi grandes que nous le voudrons; ce qui (cf l'expression de z) va donner à z des valeurs aussi grandes que nous le voudrons.

Conclusion z n'a pas de maximum.

Ceci n'est donc pas un vrai problème, puisque nous savons comment conclure dans un tel cas.

2.2 Attention Domaine dégénéré !

Considérons le programme linéaire suivant:

$$\text{Maximiser } x_2 \text{ sous les contraintes } \begin{cases} -x_1 + x_2 + x_3 = 0 \\ x_1 + x_4 = 2 \\ x_1 \geqslant 0, \ldots, x_4 \geqslant 0 \end{cases}.$$

a.

$z = x_2$

$\begin{array}{l} x_3 = x_1 - x_2 \\ x_4 = 2 - x_1 \end{array}$. Le point admissible basique est (0,0,0,2)

x_2 doit et peut entrer dans la base mais la première équation montre qu'il doit être positif (par contrainte du problème) mais majoré par 0 (sinon x_3 devient négatif); donc nous avons

b.

$z = x_1 - x_3$

$\begin{array}{l} x_2 = x_1 - x_3 \\ x_4 = 2 - x_1 \end{array}$. et nous sommes au point admissible basique (0,0,0,2), le même !!!

Nous n'avons pas bougé dans D, mais nous avons « changé de point de vue » ce qui va nous aider à continuer. L'explication est qu'en ce point il n 'y a pas que les variables hors base qui sont nulles, il y a aussi une variable de la base et alors le passage de témoin (je sors, tu entres) ne change en apparence rien.

$\mathcal{A}$ est aussi bien paramétré par $\left\{ \begin{array}{l} x_3 = 0 + x_1 - x_2 \\ x_4 = 2 - x_1 \end{array} \right.$ (et $x_1 \geqslant 0, x_2 \geqslant 0$) que par $\left\{ \begin{array}{l} x_2 = 0 + x_1 - x_3 \\ x_4 = 2 - x_1 \end{array} \right.$ (et $x_1 \geqslant 0, x_3 \geqslant 0$).

En fait, à partir du a. nous allons pouvoir continuer $\begin{array}{l} x_1 = 2 - x_4 \\ x_2 = 2 - x_3 - x_4 \\ z = 2 - x_3 - x_4 \end{array}$; et là nous savons que nous sommes arrivés au maximum.

La question qui se pose est: si un même point basique a plusieurs représentations (ce qu'on appellera un problème de dégénérescence) il y a un risque que nous entrions dans un cycle (vicieux) en passant de l'un à l'autre ....comment peut-on l'éviter, si c'est possible..

Théorème 4. *Pour éviter les « cycles vicieux » que pourrait causer un ensemble admissible dégénéré; dans ce cas on sait qu'il peut y avoir hésitation sur la variable qui doit quitter la base.*

Règle de BLAND:

i) choix de la variable entrante dans la base: prendre celle dont l'indice est le plus petit lorsqu'il y a plusieurs candidats.

ii) choix de la variable quittant la base (en cas de candidatures multiples):

Si les lignes du tableau sont comme suit

$x_{B_i} = p_i$-$\sum_{k\in N} y_{\text{ik}} x_k$, *où* $N = \{1, 2, ..., n\} \Big\backslash B$ *est égal à* $\Big\{l_1, ... l_{n-m}\Big\}, l_1 < ... < l_{n-m}$, *la variable qui va entrer dans la base est* x_v *et soit* $S = \Big\{$ *u tels que* $y_{uv} > 0 , \frac{p_u}{y_{\text{uv}}} = min\Big\{\frac{p_\alpha}{y_{\alpha v}}, \alpha \in B\Big\}\Big\}$.

Si S n'est pas réduit à un singleton on prendra la variable de plus petit indice.

A cette condition on peut montrer qu'il n'y aura pas de cycles.

2.3 Absence de valeurs initiales

Dans notre exemple qui est de taille « scolaire » il n ' y avait aucune difficulté à trouver un point basique admissible pour démarrer, mais « en vrai » ce n'est pas aussi simple; il nous faut donc une méthode qui « fournisse » automatiquement un point basique admissible « initial ».

Exemple:

Maximiser $x_1 + 2x_2$ sous les contraintes $\begin{cases} x_1 + 3x_2 + x_3 = 4 \\ 2x_2 + x_3 = 2 \\ x_1 \geqslant 0, ..., x_3 \geqslant 0 \end{cases}$.

Nous allons introduire des variables ARTIFICIELLES, auxiliaires, x_4, x_5 et étudier d'abord le programme linéaire suivant:

Maximiser $-x_4 - x_5$ sous les contraintes $\begin{cases} x_1 + 3x_2 + x_3 + x_4 = 4 \\ 2x_2 + x_3 + x_5 = 2 \\ x_1 \geqslant 0, ..., x_5 \geqslant 0 \end{cases}$;

comme x_4, x_5 sont astreints à être positifs, le maximum de la nouvelle fonction de gain sera atteint pour $x_4 = x_5 = 0$, donc la résolution du nouveau programme linéaire nous fournira des valeurs de x_1, ..., x_3 qui détermineront un point basique admissible, ce qui nous fournira des valeurs initiales pour démarrer notre résolution.

a. Résolution du programme linéaire auxiliaire:

Maximiser $-x_4 - x_5$ sous les contraintes $\begin{cases} x_1 + 3x_2 + x_3 + x_4 = 4 \\ 2x_2 + x_3 + x_5 = 2 \\ x_1 \geqslant 0, \ldots, x_5 \geqslant 0 \end{cases}$.

a. $\begin{array}{l} z = -6 + x_1 + 5x_2 + 2x_3 \\ x_4 = 4 - x_1 - 3x_2 - x_3 \\ x_5 = 2 - 2x_2 - x_3 \end{array}$.

x_1 entre dans la base , x_4 en sort

b. $\begin{array}{l} z = -2 + 2x_2 + x_3 - x_4 \\ x_1 = 4 - 3x_2 - x_3 - x_4 \\ x_5 = 2 - 2x_2 - x_3 \end{array}$.

x_3 entre dans la base , x_5 en sort

c. $\begin{array}{l} z = -x_4 - x_5 \\ x_1 = 2 - x_2 - x_4 - x_5 \\ x_3 = 2 - 2x_2 - x_5 \end{array}$.

Voilà ça y est, nous avons pour le programme auxiliaire trouvé un point admissible (2,0,2,0,0), qui nous fournit une base (1,2,3) pour ce programme, d'où une base pour le (premier) programme linéaire: (1,3).

3. Programmation linéaire et nombres entiers

Les solutions d'un problème de programmation linéaire où toutes les données sont entières sont-elles forcément des entiers ?

Exemple 5. Déterminer le maximum de z: $(x_1, x_2) \longmapsto x_1$-x$_2$

sous les contraintes $\begin{cases} x_1 + x_2 \leqslant 4 \\ 3x_1 + 5x_2 \geqslant 15 \\ x_1, x_2 \geqslant 0 \end{cases}$.

Si on trace les droites qui délimitent ce domaine on s'aperçoit que la résolution nous fournit le point (2.5,1.5) qui n'est pas à composantes entières.

Question 6. *Y a t il des problèmes de programmation linéaire où des données entières assurent des solutions entières ?*

Question 7. *Lorsque ce n'est pas le cas:*

Lorsqu'un problème de programmation linéaire est à données entières et apporte des solutions non entières, que faire ?

3.1 Matrices totalement unimodulaires

Définition 8. *Matrices totalement unimodulaires*

une matrice A à coefficients dans $\mathbb{Z}$ est dite « totalement unimodulaire » lorsque ***tous*** *ses sous-déterminants sont égaux à 1,-1 ou 0.*

(par suite ses termes appartiennent nécessairement à $\{-1, 0, +1\}$.

Théorème 9.

Les matrices extraites inversibles d'une matrice totalement unimodulaire sont à coefficients entiers.

Théorème 10. *Condition suffisante pour qu'une matrice soit totalement unimodulaire*

Une matrice A

i) à coefficients dans $\{-1, 0, +1\}$

ii) dont chaque ***colonne*** *contient* ***au plus*** *deux éléments non nuls*

iii) dont l'ensemble des ***lignes*** *peut être partitionné en deux ensembles* L_1 *et* L_2 *tels que*

- deux termes non nuls de même signe d'une même colonne doivent être l'un dans L_1 *et l'autre dans* L_2

- deux termes non nuls de signes opposés d'une même colonne doivent être tous les deux dans L_1 *ou tous les deux dans* L_2

EST TOTALEMENT UNIMODULAIRE.

Exemple 11. $\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \end{pmatrix}$

Théorème 12.

La matrice d'incidence d'un graphe bipartite est totalement unimodulaire.

Théorème 13. *Soit une matrice* ***A à coefficients entiers****, un vecteur à coefficients entiers* b *et l'ensemble convexe (polytope) défini par les contraintes* $\begin{cases} \mathrm{AX} \leqslant b \\ X \geqslant 0 \end{cases}$.

Il y a équivalence entre

i) A est totalement unimodulaire

ii) Les sommets du polytopes sont à coefficients entiers.

Donc l'application de la méthode du simplexe à un programme linéaire associé à une matrice totalement unimodulaire nous fera passer de points à coordonnées entières à points à coordonnées entières.

Définition 14. *Plus court chemin de s à t dans un réseau*

On considère un réseau, c'est à dire un graphe connexe, (S,...,C) où les arêtes sont munies de longueurs notées $\mathrm{c_{ij}}$ *et deux sommets s et t; on recherche des réels* x_{ij} *(à valeurs dans {0,1}, tels que*

$\sum_{(i,j)\in....} x_{\mathrm{ij}} c_{\mathrm{ij}}$ *soit minimale, sous les contraintes*

$$\begin{cases} \forall j \neq t, \sum_{i<j} x_{\mathrm{ij}} - \sum_{j<k} x_{\mathrm{jk}} = 0 \\ \sum_{i<t} x_{\mathrm{it}} = 1 \\ \sum_{s<k} x_{\mathrm{sk}} = 1 \end{cases}.$$

Théorème 15.

1) La recherche du plus court chemin de s à t dans le réseau ci-dessus est

le programme linéaire

Minimiser $\sum_{(i,j)\in\ldots.} x_{\mathrm{ij}} c_{\mathrm{ij}}$ *sous les contraintes* $\begin{cases} \mathrm{AX} = \begin{pmatrix} 0 \\ \ldots \\ 0 \\ 0 \\ 1 \end{pmatrix} \\ X \geqslant 0 \end{cases}$ *(où A est la matrice d'incidence du graphe privée de sa dernière ligne, car comme le graphe est connexe la matrice d'incidence est de rang n-1, si n est le nombre de sommets).*

2) Comme la matrice d'incidence est totalement unimodulaire les solutions de ce programme linéaire seront des entiers à valeurs dans {0,1}.

Définition 16. *Problème de transport*

On considère le réseau, ci-dessus, qui représentera un réseau commercial d'un grand distributeur d'un produit unique.

Le coût unitaire du transport d'un sommet i à un sommet adjacent j (=passage par l'arête (i,j)) sera représentée par c_{ij}*; certains des sommets i sont munis d'une valeur notée* b_i*, positive ou négative; lorsque* b_i*<0 nous considérerons que le sommet i dispose d'un stock |*b_i*| qu'il peut fournir, lorsque* b_i*>0 nous considérerons que le sommet i désire recevoir la quantité* b_i*.*

On supposera pour simplifier que $\sum_{i\in S} b_i = 0$ *(quitte à ajouter un sommet factice).*

On cherche à faire parvenir les commandes de telle sorte que toutes les commandes soient satisfaites et que le coût du transport soit minimal.

Théorème 17.

1) La recherche du plan de transport le plus économique dans le réseau ci-dessus est

le programme linéaire

Minimiser $\sum_{(i,j)\in\ldots.} x_{\mathrm{ij}} c_{\mathrm{ij}}$ *sous les contraintes* $\left\{\begin{array}{l}\mathrm{AX}=b\\ X \geqslant 0\end{array}\right.$ *(où A est la matrice d'incidence du graphe).*

2) Comme la matrice d'incidence est totalement unimodulaire les solutions de ce programme linéaire seront des entiers.

D'où, à nouveau en faisant appel à la dualité, un algorithme spécifique comme celui du marche-pied (stepping stone), dont l'énoncé et la justification peuvent cependant se donner sans expression de dualité.

Remarque 18.

Le problème de transport est un programme linéaire, il peut donc être résolu par la méthode du simplexe; cependant il est possible d'appliquer une méthode de simplexe adaptée aux problèmes de réseau (out-of-kilter).

Il en est de même pour le problème d'affectation optimale, dont nous avons vu qu'il est un cas particulier de ce problème.

4. Travaux dirigés

Exercice 1.

Résoudre en utilisant le tableau du simplexe

Maximiser f:$(x1, x2) \longmapsto x1 + 2x2$

sous les contraintes $\left\{\begin{array}{l}\text{-}x1 + 2x2 \leqslant 4\\ \text{-x1+x2}\leqslant 1\\ x1\text{-4x2}\leqslant 4\\ x1 \geqslant 0, x2 \geqslant 0\\ \end{array}\right.$.

Exercice 2. Résoudre en utilisant le tableau du simplexe (cyclage)

Maximiser f:$(x1, x2, x3) \longmapsto x1 + x2 + x3$

sous les contraintes $\begin{cases} x1 + x2 + x3 \leqslant 10 \\ 4x1+x2+2x3 \leqslant 24 \\ 4x1 + 5x2+8x3 \leqslant 60 \\ 2x1 + x2 + 4x3 \leqslant 24 \\ x1 \geqslant 0, x2 \geqslant 0, x3 \geqslant 0, x4 \geqslant 0 \end{cases}$

Exercice 3.

Une PME peut produire avec une machine donnée, qui travaille 45 heures par semaine , trois types de produits P1,P2,P3.

Les bénéfices unitaires fournis par la production de ces produits sont:

P1: 4 Euros

P2: 12 Euros

P3: 3 Euros

La machine peut produire par heure

50 éléments de type P1

ou 25 éléments de type P2

ou 75 éléments de type P3

Une étude de marché établit que la demande hebdomadaire de produits

de type P1 ne dépasse pas 1000

de type P2 ne dépasse pas 500

de type P3 ne dépasse pas 1500

Déterminer la meilleure répartition du temps de travail -machine hebdomadaire.

Exercice 4. Problème de transport

Soit le problème de transport résumé dans le tableau suivant

	$C1$	$C2$	$C3$	
$D1$	5	3	4	8
$D2$	6	7	2	9
	4	5	6	

1. Représenter le problème sous la forme d'un programme linéaire où on précisera

les inconnues, les contraintes.

attention : on veut minimiser un coût et pas maximiser un gain!!!

2. Expliquer pourquoi la résolution du programme linéaire nous fournira des solutions entières (donc utilisables)

3. Résoudre.

Remarque 19.

- Si T(B) est un tableau du simplexe tels que les coefficients des variables non basiques sont positifs (ou nuls) (c'est à dire pour tout k, $(z_k - c_k)$ $\geqslant$0) alors la solution basique admissible associée à B est optimale;

- Une variable x_v non basique peut entrer dans la base si et seulement si son coefficient dans la première ligne du tableau (c'est à dire « $(z_k - c_k)$ » dans la ligne de z, le gain) est strictement négatif dans le cas où il y a plusieurs candidats il existe plusieurs stratégies:

i) prendre celui dont la valeur absolue du coefficient est la plus grande.

ii) prendre celui dont l'indice est le plus faible.

- La variable basique x_u qui quittera la base sera celle dont la contrainte de positivité limite le plus la croissance de la variable entrante x_v ; si on écrit les lignes du tableau comme suit

$x_{B_i} = p_i$-$\sum_{k\in N} y_{\text{ik}} x_k$, où $N = \{1, 2, ..., n\} \Big\backslash B$ est égal à $\Big\{l_1, ...l_{n-m}\Big\}$, $l_1 < ... < l_{n-m}$, on sélectionnera les α tels que $y_{\alpha v} > 0$ et on prendra u l'élément de N tel que $\frac{p_u}{y_{\text{uv}}}$= $\min\Big\{\frac{p_\alpha}{y_{\alpha v}}, \alpha \in B \text{ tq } y_{\alpha v} > 0\Big\}$.

Si un tel u n'est pas unique voir plus bas la règle de Bland.

5. Programme linéaire sous forme canonique

On dira que le programme linéaire est sous forme canonique se présente comme suit:

Maximiser ^{t}cX sous les contraintes $\left\{\begin{array}{l}\text{AX} \leqslant b\\ X \geqslant 0\end{array}\right.$, où on comprendra que AX$\leqslant b$ signifie $\forall i \in \{1, ..., m\}, \sum_{j=1}^n a_{\text{ij}} x_j \leqslant b_i$.

Théorème 20.

*i) On peut transformer un programme linéaire sous forme canonique en un programme linéaire sous forme standard **équivalent** en ajoutant m variables D'ECART auxiliaires comme suit*

pour chaque $i \in \{1, ..., m\}$ $\sum_{j=1}^n a_{\text{ij}} x_j \leqslant b_i \Longleftrightarrow \exists y_i \geqslant 0, \sum_{j=1}^n a_{\text{ij}} x_j + y_i = b_i$.

ii) Si le programme originel (canonique) possède une solution le programme standard associé possède une solution (basique); la réciproque est évidente.

6. Complexité

Si la pratique montre une complexité polynomiale ($\mathrm{O}(\mathrm{mn}^3)$) il existe des exemples de complexité exponentielle:

Exemple 21. Maximiser $\sum_{i=1}^{n} x_i$

sous la contrainte $\begin{cases} x_1 \leqslant 1 \\ 2x_1 + x_2 \leqslant 3 \\ 2x_1 + 2x_2 + x_3 \leqslant 9 \\ \ldots \\ \ldots \\ 2x_1 + \ldots. + 2x_{n-1} + x_n \leqslant 3^{n-1} \\ x_1 \geqslant 0, \ldots, x_n \geqslant 0 \end{cases}$. L'application de la méthode du simplexe contraindra à traverser les 2^n sommets du polyèdre des points admissibles; ce qui signifie une complexité exponentielle.

Il existe d'autres méthodes de résolution des programmes linéaires, de complexité polynomiale, comme celle de l'ellipsoïde (Shor, Khachiyan,Yudin) qui date des années 70 (plus ou moins abandonnée car peu efficace) ou celle des points intérieurs (Karmarkar) qui date des années 80, considérée comme prometteuse, dont la complexité est polynomiale ($O\big(n^{7/2}\big)$).

Objectifs:

1. Savoir reconnaître un progamme linéaire sous forme standard (=équationnelle).

2. Comprendre la nature de l'ensemble des points admissibles.

3. Comprendre les principes de la méthode du simplexe, savoir ce qu'est une base admissible, pourquoi et comment une variable entre dans la base, pourquoi et quelle variable sort de la base.

4. Savoir construire le tableau du simplexe.

5. Savoir comment on reconnaît l'optimalité d'une solution.

6. Savoir comment on reconnaît que le problème ne possède pas de solution.

7. Savoir appliquer la méthode du simplexe à partir d'une solution initiale admissible.

8. Savoir introduire des variable auxiliaires afin d'obtenir une solution initiale admissible.

9. Connaître la règle de Bland pour éviter les « cycles vicieux ».

10. Savoir comment traiter le cas d'un programme linéaire sous forme inéquationnelle (=canonique).