Programmation linéaire méthode de simplex

L'objectif de cette cour est utilisation de la méthode de simplex pour résoudre un problème optimisation afin de Calculer le max(z) d'une fonction linéaire de la forme :

$$\max(Z)=\sum_{I=1}^{N} C_I.x_I = C_1.x_1 + C_2.x_2 + ... + C_N.x_N$$

Avec les contraintes de la forme :

$A_{(I,J)}.x_J \le b_I \quad \rightarrow I=1:M \quad ,J=1:N$

Tel que **N** Nombre de variable et ***M*** Nombre des contraintes Avec $x_J > 0$

$$\begin{cases} A_{(1,1)}.x_1 + A_{(1,2)}.x_2 + ... + A_{(1,N)}.x_N \le b_1 \\ A_{(2,1)}.x_1 + A_{(2,2)}.x_2 + ... + A_{(2,N)}.x_N \le b_2 \\ . \\ . \\ A_{(M,1)}.x_1 + A_{(M,2)}.x_2 + ... + A_{(M,N)}.x_N \le b_M \end{cases}$$

Les étapes des calculs de la méthode de simplex :

- **écrire le système sous forme standard**

Il s'agit convertir le problème établi sous forme canonique (système d'inéquation) sous la forme standard (système d'équation avec variable d'écarts). Les variables d'écarts introduites au cours de cette transformation représentent les contraintes techniques et commerciales disponibles qu'il convient de saturer.

Forme canonique $\qquad\qquad$ Forme standard

$$\begin{cases} A_{(1,1)}.x_1 + A_{(1,2)}.x_2 + ... + A_{(1,N)}.x_N \le b_1 \\ A_{(2,1)}.x_1 + A_{(2,2)}.x_2 + ... + A_{(2,N)}.x_N \le b_2 \\ . \\ . \\ A_{(M,1)}.x_1 + A_{(M,2)}.x_2 + ... + A_{(M,N)}.x_N \le b_M \end{cases} \rightarrow \begin{cases} A_{(1,1)}.x_1 + A_{(1,2)}.x_2 + ... + A_{(1,N)}.x_N + x_{N+1} = b_1 \\ A_{(2,1)}.x_1 + A_{(2,2)}.x_2 + ... + A_{(2,N)}.x_N + x_{N+2} = b_2 \\ . \\ . \\ A_{(M,1)}.x_1 + A_{(M,2)}.x_2 + ... + A_{(M,N)}.x_N + x_{N+M} = b_M \end{cases}$$

- Construire la matrice augmentée correspondant à la forme standard : $[AG]=\begin{bmatrix} A & | & I_n & | & b \end{bmatrix}$
- Initialisation de vecteur de la base : `jbase=[n+1:nm]'`

$$[AG]=\left(\begin{array}{cccc|cccc|c} A_{1,1} & A_{1,2} & . & A_{1,N} & 1 & 0 & . & 0 & b_1 \\ A_{2,1} & A_{2,2} & & . & 0 & 1 & & . & b_2 \\ . & & . & . & . & & 1 & . & . \\ A_{M,1} & A_{M,2} & . & A_{M,N} & 0 & 0 & . & 1 & b_M \end{array}\right)$$

$$[CG]=\left[\begin{array}{cccc|cccc|c} C_1 & C_2 & . & C_N & 0 & 0 & . & 0 & 0 \end{array}\right]$$

- On cherche le max et sa position ***jmax*** de tableau **CG :** **`[max_cg, jmax]=max(cg(1:nm))`**
- On calcul le rapport entre les deux colon : **`b_a=ag(:,nm+1)./ag(:,jmax)`**
  - On cherche le min et sa position ***imin*** de tableau ***`b_a>0`*:** `[min_b,imin]=`**`min`**`(b_a)`
- On désigne le pivot comme suite : `pivot=ag(`***`imin`***`,`***`jmax`***`)`
  - On désigne la ligne de pivot comme suite : `lin_p=ag(`***`imin`***`,:)`
  - Indication les variables de la base : **jbase(imin)=jmax**
  - on effectue des opérations élémentaire pour chaque lignes de la matrice **`ag(i,j)`** :

```
          ag(imin,:) = lin_p(:)/pivot   ;
          i=1 ;
         while( i<=m )
               if(  i~=imin )
   r=ag(i,jmax) /pivot ;    ag(i,:)=ag(i,:)- lin_p(:)' * r ;
               end
            i=i+1 ;
         end
```

- même chose pour la ligne des coefficients **cg(j):**

```
r=  cg(jmax)   /pivot  ;
cg(:)= cg(:)  - lin_p(:)* r ;
```

Si les coefficient Cg sont nuls ou négatifs donc on a un optimum qui est : **`zmax=Cg(nm+1)`**
Si non on répète les même étapes jusqu'à ***`max(cg)<=0`***

Exemple 1:

$$\max(z) = 4.x_1 + 5.x_2$$

$$\begin{cases} 2.x_1 + 1.x_2 \le 8 \\ 1.x_1 + 2.x_2 \le 7 \\ 0.x_1 + 1.x_2 \le 4 \end{cases} \rightarrow \begin{cases} 2.x_1 + 1.x_2 + 1.x_3 = 8 \\ 1.x_1 + 2.x_2 + 1.x_4 = 7 \\ 0.x_1 + 1.x_2 + 1.x_5 = 4 \end{cases}$$

$Telque: \quad x_1 \ge 0,\ x_2 \ge 0$

	X1	X2	X3	X4	X5	b
X3	2	1	1	0	0	8
X4	1	2	0	1	0	7
X5	0	1	0	0	1	4
CG	4	**5**	0	0	0	/
		Jmax=2				

1ère étape : jbase = 3 4 5

Max(CG)= 5 ; jmax=2 ; B_A=b(:)/a(: ,jmax=2)= $\begin{pmatrix} 8/1 \\ 7/2 \\ 3/1 \end{pmatrix} = \begin{pmatrix} 8 \\ 3.5 \\ 4 \end{pmatrix}$ **Imin=2**

Min(B_A)=3.5>0 ; imin= 2 ; pivot=ag(imin=2 ,jmax=2)=2 ; jbas(imin=2)=jmax=2
Lin_p(:)= ag (imin=2, :)=(1 2 0 1 0 7)
ag (imin=2, :)=Lin_p(:)/pivot=(1 2 0 1 0 7)/2=(0.5 1 0 0.5 0 3.5)
pour $i \ne i\min$ on a r=ag(i,jmax) /pivot ; ag(i,:)=ag(i,:)- lin_p(:)' * r ;
Pour i=1 on a r= ag(i=1,jmax=2) /pivot=1/2
ag(1,:)=ag(1,:)- lin_p(:)' * r =(2 1 1 0 0 8)-(1 2 0 1 0 7)*1/2
=(1.5 0 1 -0.5 0 4.5)
Pour i=3 on a r= ag(i=3,jmax=2) /pivot=1/2
ag(3,:)=ag(3,:)- lin_p(:)' * r =(0 1 0 0 1 4)-(1 2 0 1 0 7)*1/2
=(0.5 0 0 0.5 1 0.5)
r=cg(jmax=2)/pivot=5/2;
cg(:)= cg(:) - lin_p(:)* r =(4 5 0 0 0 0)-(1 2 0 1 0 7)*5/2
=(1.5 0 0 -2.5 0 -17.5)

//

2ème étape
jbase = 3 4 2

	X1	X2	X3	X4	X5	b
X3	1.5	0	1	-0.5	0	4.5
X2	0.5	1	0	0.5	0	3.5
X5	-0.5	0	0	-0.5	1	0.5
CG	**1.5**	0	0	-2.5	0	-17.5
	Jmax=1					

Max(CG)= 4 ; jmax=1 ;

B_A=b/a(: ,jmax=1)= $\begin{pmatrix} 4.5/1.5 \\ 3.5/0.5 \\ 0.5/0.5 \end{pmatrix} = \begin{pmatrix} 3 \\ 7 \\ -1 \end{pmatrix}$ **Imin=1**

Min(B_A)=3>0 ; imin= 1 ; pivot= ag (imin=1,jmax=1)=1.5 ; jbas(imin=1)=jmax=1
Lin_p(:)=ag(imin=1, :)=(1.5 0 1 -0.5 0 4.5)
ag (imin=1, :)=Lin_p(:)/pivot=(1.5 0 1 -0.5 0 4.5) /1.5=(1 0 0.6667 -0.3333 0 3)
pour $i \ne i\min$ r=ag(i,jmax) /pivot ; ag(i,:)=ag(i,:)- lin_p(:)' * r ;
Pour i=2 ; r= ag(i=2,jmax=1) /pivot=0.5/1.5 =1/3 ;
ag(2,:)=ag(2,:)- lin_p(:)' * r =(0.5 1 0 0.5 0 3.5)-(1.5 0 1 -0.5 0 4.5)*0.5/1.5
=(0 1 -0.33 0.67 0 2)

Pour i=3 ; r= ag(i=3,jmax=1) /pivot=-0.5 /1.5 =-1/3 ;
ag(3,:)=ag(3,:)- lin_p(:)' * r =(-0.5 0 0 -0.5 1 0.5)-(1.5 0 1 -0.5 0 4.5)*(-1/3)
=(0 0 0.33 -0.67 1 2)

r=cg(jmax=1)/pivot=1.5/1.5=1 ;
cg(:)= cg(:) - lin_p(:)* r =(1.5 0 0 -2.5 0 -17.5)-(1.5 0 1 -0.5 0 4.5)*1
=(0 0 -1 -2 0 -22)

	X1	X2	X3	X4	X5	b
X1	1	0	0.66	-0.33	0	**3**
X2	0	1	-0.33	0.67	0	**2**
X5	0	0	0.33	-0.67	1	2
CG	0	0	-1	-2	0	**-22**

Toutes les valeurs CG sont négatives donc on a un optimum zmax = abs(-22)=22 dans le point x1=3 , x2=2

Présentation graphique

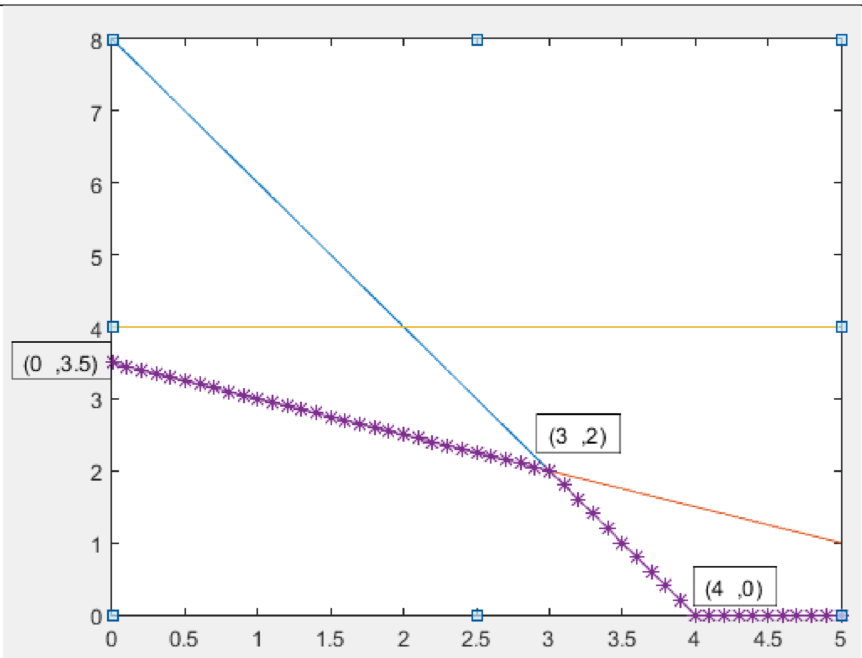
$$\max(z) = 4.x_1 + 5.x_2$$

$$\begin{cases} 2.x_1 + 1.x_2 \leq 8 \\ 1.x_1 + 2.x_2 \leq 7 \\ 0.x_1 + 1.x_2 \leq 4 \end{cases}$$

$$Telque: \quad x_1 > 0,\ x_2 > 0$$

X1	X2	Z=4.x1+5.x2
0	3.5	14
3	**2**	**22**
4	0	16

Pour x1=3 ; x2=2 on a Zmax=22



$$\% \ exemple\, 1$$
$$\max(z) = 4.x_1 + 5.x_2$$
$$\begin{cases} 2.x_1 + 1.x_2 \le 8 \\ 1.x_1 + 2.x_2 \le 7 \\ 0.x_1 + 1.x_2 \le 4 \end{cases}$$
$$Telque: \quad x_1 > 0,\ x_2 > 0$$
$$\% \ exemple\, 2$$
$$\max(z) = 14.x_1 + 15.x_2 + 24.x_3$$
$$\begin{cases} 4.x_1 + 1.x_2 + 1.x_3 \le 260 \\ 1.x_1 + 2.x_2 + 2.x_3 \le 200 \\ 3.x_1 + 3.x_2 + 4.x_3 \le 380 \end{cases}$$
$$Telque: \quad x_1 > 0,\ x_2 > 0,\ x_3 > 0$$

```
clc  ;  clear all
% example 1
a=[2   1  ; 1  2 ;  0  1];
b=[8;  7 ; 4]    ;      c=[4 5] ;
% example 2
%a=[4   1   1; 1  2   2;  3   3   4];
%b=[260;  200 ; 380] ;   c=[14 15 24];
 [m,n]=size(a)  ; nm=n+m  ;
ag=[a  ,  eye(m)    , b]
cg=[c  , zeros(1,m) , 0]
jbase=[n+1:nm]'
%=========================
while(max(cg(1:nm)) >0)
      [max_cg, jmax]=max(cg(1:nm));
       b_a=ag(:,nm+1)./ag(:,jmax) ;
       min_b=b_a(1) ; imin=1 ;
      for i=2:m
          if(b_a(i)<min_b & b_a(i)>0)
            min_b=b_a(i) ; imin=i;
          end
      end
      pivot=ag(imin,jmax)   ;      lin_p=ag(imin, :) ;
      jbase(imin)=jmax ;
      ag(imin,:) = lin_p(:)/pivot  ;
      i=1 ;
     while( i<=m )
        if(i~=imin )
   r=ag(i,jmax) /pivot ;  ag(i,:)=ag(i,:)- lin_p(:)' * r ;
        end
        i=i+1 ;
     end
  r=  cg(jmax)/pivot  ;  cg(:)= cg(:)  - lin_p(:)* r ;
   ag
   cg
   jbase
end

zmax= abs(cg(nm+1))
jbase
b=ag(:,nm+1)
```

Exercice : **Programmation linéaire méthode de simplex**

Soit une entreprise qui fabrique deux types de camions (types A et B). Cette entreprise est divisée en trois ateliers. Ateliers I fabriquant les moteurs, ateliers II fabriquant les carrosseries, ateliers III étant chargé de l'assemblage.

Les temps unitaires pour chacune des trois opérations et chaque de camions sont consignés dans le tableau suivant :

ateliers	Camions type A	Camions type B
I moteurs	1 H	3 H
II carrosseries	2 H	1 H
III assemblage	1 H	1 H

Par ailleurs, l'étude des capacités de production des 3 ateliers a dégagé qu'en un mois, **450** heures de travail pouvaient être utilisées dans l'atelier I, **350** heures dans l'atelier II, **200** dans l'atelier III.

Enfin, on sait que le bénéfice unitaire réalisé par l'entreprise sur les camions de type A s'élève à **4000** et que celui réalise sur les camions de type B est de **8000**.

La question que l'on se pose est la suivant : quel doit être la production mensuelle en camions de chaque type pour rendre le bénéfice de l'entreprise le plus grand possible ?

Nous pouvons formaliser ce problème de la façon suivant :

Soit x_1 la production mensuelle en camions de type A et x_2 celle en camions de type B.

Les contraintes de disponibilité d'heures de travail dans chacun des ateliers peuvent s'écrire :

$$\begin{cases} \text{Atelier I} \;\; : \; 1.x_1 + 3.x_2 \leq 450 \\ \text{Atelier II} \; : \; 2.x_1 + 1.x_2 \leq 350 \\ \text{Atelier III:} \; 1.x_1 + 1.x_2 \leq 200 \end{cases} \qquad \text{(a)}$$

Par ailleurs, on a évidemment $x_1 \; et \; x_2 \geq 0$ (b)

Enfin, il s'agit de trouver $\boldsymbol{x_1}$ et $\boldsymbol{x_2}$ répondant aux contraintes (a) et (b) et rendant la fonction $\boldsymbol{Z=4000.x_1+8000.x_2}$ est maximale.

Ce problème constitue un programme linéaire.

Dans le cas à deux variables, on peut résoudre le problème en utilisant une représentation géométrique.

D'une façon générale, un programme linéaire consiste à maximiser (ou minimiser) une fonction linéaire de n variable, ces variable étant assujetties à respecter un ensemble de m contraintes également linéaire.

Exercice : **Programmation linéaire méthode de simplex**

Soit une entreprise qui fabrique deux types de camions (types A et B). Cette entreprise est divisée en trois ateliers. Ateliers I fabriquant les moteurs, ateliers II fabriquant les carrosseries, ateliers III étant chargé de l'assemblage.

Les temps unitaires pour chacune des trois opérations et chaque de camions sont consignés dans le tableau suivant :

ateliers	Camions type A	Camions type B
I moteurs	1 H	3 H
II carrosseries	2 H	1 H
III assemblage	1 H	1 H

Par ailleurs, l'étude des capacités de production des 3 ateliers a dégagé qu'en un mois, **450** heures de travail pouvaient être utilisées dans l'atelier I, **350** heures dans l'atelier II, **200** dans l'atelier III.

Enfin, on sait que le bénéfice unitaire réalisé par l'entreprise sur les camions de type A s'élève à **4000** et que celui réalise sur les camions de type B est de **8000**.

La question que l'on se pose est la suivant : quel doit être la production mensuelle en camions de chaque type pour rendre le bénéfice de l'entreprise le plus grand possible ?

Nous pouvons formaliser ce problème de la façon suivant :

Soit x_1 la production mensuelle en camions de type A et x_2 celle en camions de type B.

Les contraintes de disponibilité d'heures de travail dans chacun des ateliers peuvent s'écrire :

$$\begin{cases} \text{Atelier I} \ : 1.x_1 + 3.x_2 \leq 450 \\ \text{Atelier II} \ : 2.x_1 + 1.x_2 \leq 350 \\ \text{Atelier III:} \ 1.x_1 + 1.x_2 \leq 200 \end{cases} \qquad \text{(a)}$$

Par ailleurs, on a évidemment $x_1 \; et \; x_2 \geq 0$ (b)

Enfin, il s'agit de trouver x_1 et x_2 répondant aux contraintes (a) et (b) et rendant la fonction $Z=4000.x_1+8000.x_2$ est maximale.

Ce problème constitue un programme linéaire.

Dans le cas à deux variables, on peut résoudre le problème en utilisant une représentation géométrique.

D'une façon générale, un programme linéaire consiste à maximiser (ou minimiser) une fonction linéaire de n variable, ces variable étant assujetties à respecter un ensemble de m contraintes également linéaire.