**R4.04 : Méthodes d’optimisation**

**TD 1**

**Partie 1 : Formulation et résolution d’un programme linéaire**

**Exercice 1 :**

Vous êtes en charge de la gestion de production d'une entreprise de fabrication de meubles. Vous disposez de deux types de matières premières : le bois et le métal. Vous pouvez utiliser ces matières premières pour produire deux types de meubles : des chaises et des tables. Chaque chaise nécessite 2 unités de bois et 3 unités de métal, et chaque table nécessite 5 unités de bois et 4 unités de métal. Vous avez un stock limité de bois (180 unités) et de métal (200 unités) disponibles pour produire des meubles. De plus, vous avez un nombre limité d'heures de travail disponibles pour produire des meubles (80 heures). Il est nécessaire 1 heure pour produire une chaise et 2 heures pour produire une table.

Enfin, chaque chaise rapporte un bénéfice de 15€ et chaque table rapporte un bénéfice de 20€.

Votre objectif est de maximiser les bénéfices en produisant le nombre optimal de chaises et de tables.

1. Modélisez mathématiquement le problème de production sous forme de programme linéaire.
2. Résolvez le modèle graphiquement en utilisant les techniques de programmation linéaire.
3. Résolvez le modèle en utilisant l'algorithme simplex en ligne.
4. L’entreprise souhaite réduire le nombre d’heures de travail à 75 heures. Reformulez le problème en prenant en compte la nouvelle contrainte.
5. Est-ce que le nouveau problème peut être résolu graphiquement ? pourquoi ?
6. Quelle méthode peut-on utiliser dans ce cas ?
7. Résoudre ce problème graphiquement en utilisant l’algorithme Branch and Bound
8. Donnez la solution optimale ? Conclure
9. Peut-on reformuler le programme afin de déterminer le nombre minimale d’heures de travail à effectuer pour réaliser encore le même gain.

**Correction :**

Voici une façon de résoudre le problème de gestion de production présenté ci-dessus :

1. Modélisation mathématique : Le modèle mathématique pour maximiser les bénéfices peut être exprimé de la manière suivante :

Max Z = 15x + 20y

Sujet aux contraintes :

x + 2y <= 80 (nombre limité d'heures de travail disponibles)

3x + 4y <= 200 (stock limité de métal)

2x + 5y <= 180 (stock limité de bois)

x, y >= 0 (nombre de chaises et de tables produites doit être positif)

1. Résolution graphique :



1. Résolution avec Simplex : On peut résoudre le modèle en utilisant l'algorithme simplex. L'algorithme consiste à choisir une variable de base, à calculer les ratios et à sélectionner la variable sortante. Le processus est répété jusqu'à ce qu'on obtienne une solution optimale. En utilisant cette méthode, on obtient également la solution optimale x=40, y=20, Z= 1000

Donc en résolvant le problème de gestion de production, la solution optimale est de produire 40 chaises et 20 tables pour un bénéfice total de 1000€.

1. Nouveau problème :

Max Z = 15x + 20y

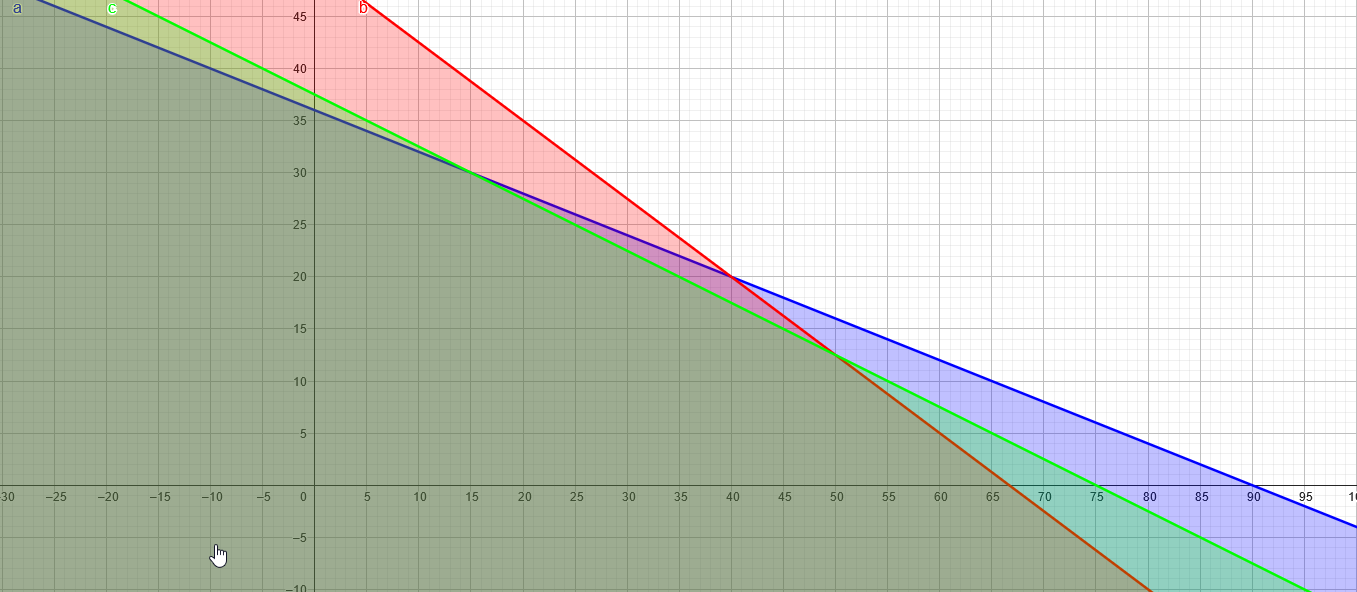
Sujet aux contraintes :

x + 2y <= 75 (nombre limité d'heures de travail disponibles)

3x + 4y <= 200 (stock limité de métal)

2x + 5y <= 180 (stock limité de bois)

x, y >= 0 (nombre de chaises et de tables produites doit être positif)



Solution graphique :

X=37.5 et y= 25

Solution par branch and bound :

X=52 et Y=11 et Z=1000

1. Notre nouveau programme linéaire serait de la forme suivante:

Minimize: z= x1 + 2x2 (contrainte d'heures de travail)

Sous contraintes :

2x1 + 5x2 <= 180 (contrainte de disponibilité de bois)

3x1 + 4x2 <= 200 (contrainte de disponibilité de métal)

15x1+20x2 >=1000 (contrainte sur le bénéfice)

x1,x2 >= 0 (contrainte de non-négativité)

Solution :

x = 64.0, y = 2.0 => Z = 68.0

Programme python fournit sur le fichier joint.

**Partie 2 : Programmation linéaire avec simplexe :**

1. Soit x le nombre d'unités de lampes de type SP-100 fabriquées par jour, et y le nombre d'unités de lampes de type SP-200 fabriquées par jour.

La fonction objectif à maximiser est la contribution au bénéfice :

**Maximiser Z= 15 x + 22 y**

Sous les contraintes suivantes :

Si toutes les lampes étaient du type SP-100, le département pourrait en fabriquer 60 unités par jour. Le temps de fabrication pour le type SP-200 est le double de temps de fabrication pour le type SP-100. Donc, la première contrainte est comme suit :

x + 2y ≤ 30

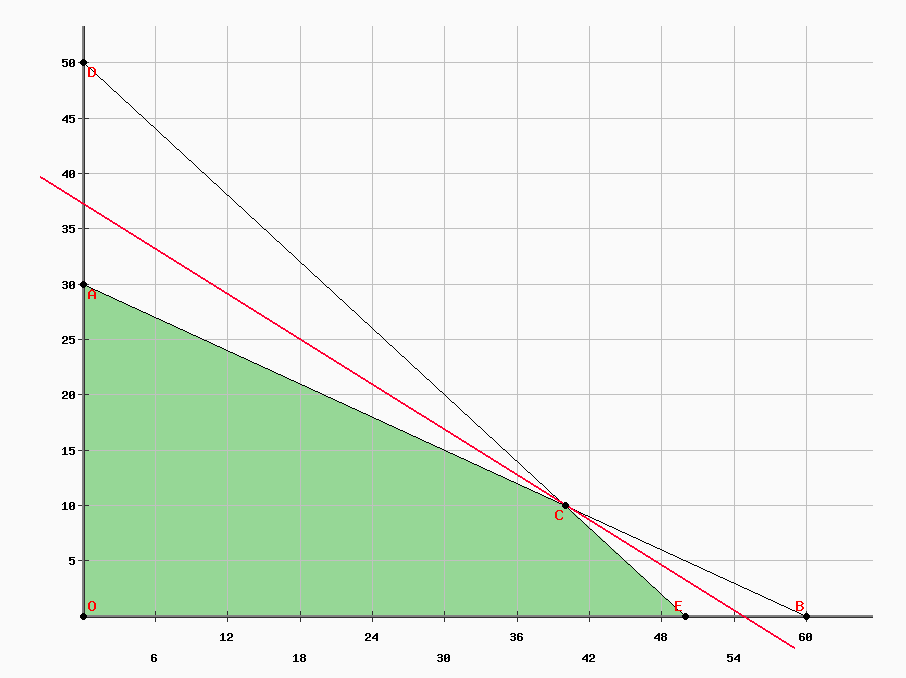
La quantité totale des deux types de lampes ne peut excéder 50 unités par jour :

x + y ≤ 50

Les variables de décision doivent être non négatives :

x ≥ 0, y ≥ 0

1. Le problème formulé est linéaire car **la fonction objectif** et toutes les **contraintes** sont des fonctions **linéaires** des variables de décision **positives**.
2. La zone des solutions réalisables est représentée par la zone en vert sur la figure suivante :



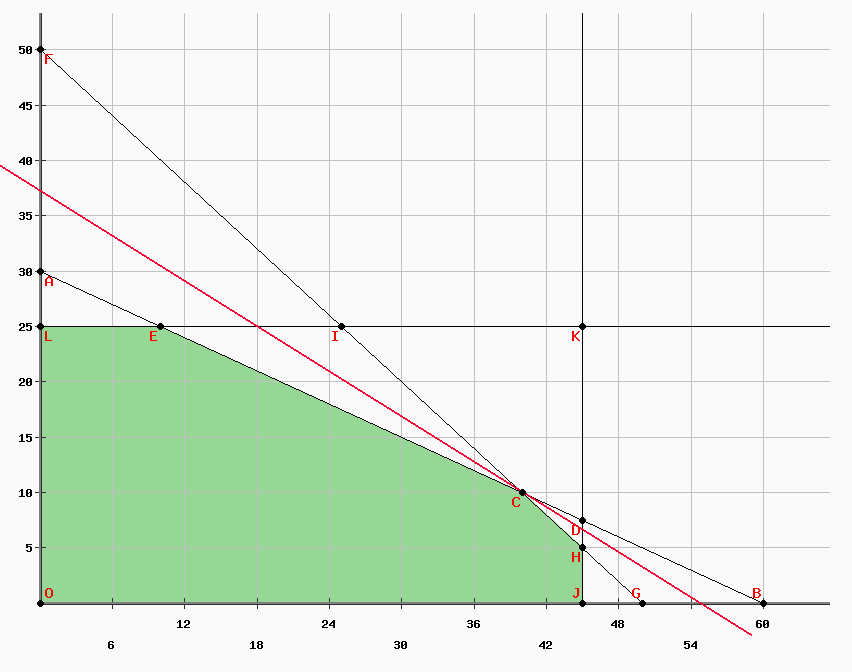
1. Les solutions aux sommets de la zone des solutions réalisables sont données par le tableau suivant :

|  |  |  |  |
| --- | --- | --- | --- |
| **Point** | **Coordonée X (X1)** | **Coordonée Y (X2)** | **Valeur de la fonction (Z)** |
| O | 0 | 0 | 0 |
| A | 0 | 30 | 660 |
| C | 40 | 10 | 820 |
| E | 50 | 0 | 750 |

1. La solution optimale est donnée par le point C du tableau précédent car c’est le point qui assure la meilleure valeur de la fonction objectif.
2. L’entreprise utilise des filaments spécifiques (un pour chaque type de lampes) pour la fabrication de ces lampes et qui ont un taux élevé de rejet et la quantité maximale disponible pour le type SP-100 est de 45 filaments par jour et de 25 par jour pour le type SP-200. Alors, deux contraintes s’ajouteront à notre problème à savoir :

X <= 45 et y <= 25

1. La zone des solutions réalisables est représentée par la zone en vert sur la figure suivante :



Les solutions aux sommets de la zone des solutions réalisables sont données par le tableau suivant :

|  |  |  |  |
| --- | --- | --- | --- |
| **Point** | **Coordonée X (X1)** | **Coordonée Y (X2)** | **Valeur de la fonction (Z)** |
| O | 0 | 0 | 0 |
| C | 40 | 10 | 820 |
| E | 10 | 25 | 700 |
| H | 45 | 5 | 785 |
| J | 45 | 0 | 675 |
| L | 0 | 25 | 550 |

La solution optimale est donnée par le point C du tableau précédent car c’est le point qui assure la meilleure valeur de la fonction objectif.

**Si par contre, on modifie la condition de la question 6 à x<=35 (on a uniquement 35 filaments pour fabriquer les lampes de type SP-100), la solution optimale précédente n’est plus d’actualité car elle ne vérifie plus les contraintes du programme linéaire. Alors, graphiquement on obtient une solution optimale avec x=35 et y=12.5. Forcément, cette solution n’est pas acceptable car il nous faut des valeurs entières pour les variables de décision.**

**En utilisant le programme Python suivant :**

**from** pulp **import** **\***

# Create the problem variable

prob **=** LpProblem**(**"Meubles production"**,** LpMaximize**)**

# Create problem variables

x **=** LpVariable**(**"x"**,** 0**,** **None,** cat**=**'Integer'**)** # Nombre de lampes SP-100

y **=** LpVariable**(**"y"**,** 0**,** **None,** cat**=**'Integer'**)** # Nombre de lampes SP-200

# Add objective function

prob **+=** 15**\***x **+** 20**\***y**,** "Z"

# Add constraints

prob **+=** x **+** 2**\***y **<=** 60 # nombre limité d'heures de travail disponibles

prob **+=** x **+** y **<=** 50 # nombre max de lampes à produire

prob **+=** x **<=** 35

prob **+=** y **<=** 25

# Solve the problem

status **=** prob**.**solve**()**

# Print the solution

**print(**f'Status: {LpStatus**[**status**]**}'**)**

**print(**f'x = {x**.**value**()**}, y = {y**.**value**()**} => Z = {value**(**prob**.**objective**)**}'**)**

**Partie 2 : Programmation non linéaire avec simplexe :**

1. On peut modéliser ce problème en utilisant la programmation linéaire. Les variables décisionnelles sont le nombre de chaises (x1) et le nombre de tables (x2) produits, qui sont soumis aux contraintes suivantes :

* Contraintes de matières premières : 2x1 + 3x2 <= 180 (nombre limité de bois disponible) x1 + 2x2 <= 120 (nombre limité de métal disponible)
* Contraintes de temps de travail : 2x1 + 3x2 <= 80 (nombre limité d'heures de travail disponible)
* Contraintes de nombre de chaises et de tables produits : x1 >= 0, x2 >= 0 (nombre de chaises et de tables produits doit être positif)

La fonction objectif est de maximiser le bénéfice net, qui est la différence entre le bénéfice total et les coûts de matières premières : Z = 15x1 + 20x2 – 2(2x1+3x2) – 3(x1+2x2)

Z = 8x1 + 8x2

La forme finale de la modélisation mathématique est donc :

Maximiser Z = 8x1 + 8x2

Sujet aux contraintes :

2x1 + 3x2 <= 180

x1 + 2x2 <= 120

2x1 + 3x2 <=80

x1 >= 0, x2 >= 0

1. **Solution avec simplexe :**

**from** scipy**.**optimize **import** linprog

c **=** **[-**8**,** **-**8**]**

A **=** **[[**2**,** 3**],** **[**1**,** 2**],** **[**2**,** 3**]]**

b **=** **[**180**,** 120**,** 80**]**

x0\_bounds **=** **(**0**,** **None)**

x1\_bounds **=** **(**0**,** **None)**

res **=** linprog**(**c**,** A\_ub**=**A**,** b\_ub**=**b**,** bounds**=[**x0\_bounds**,** x1\_bounds**],** method**=**'simplex'**)**

**print(**"Optimal value:"**,** res**.**fun**,** "\nX:"**,** res**.**x**)**

**La solution optimale obtenue avec ce programme est x=40, y=0 et un bénéfice à 320.**

1. Si on utilisait les fonctions de coûts non-linéaires fournit dans l’exercice :

Le coût du bois suit la fonction suivante :

Le coût du métal suit la fonction suivante :

Ce qui change dans la formulation du problème c’est la fonction objective. La nouvelle fonction objectif est commen suit :

Z = 15x1 + 20x2 -

Cette fonction n’est plus linéaire. Donc, notre proglramme n’est pas linaire. C’est un problème **d’optimisation quadratique**. Pour résoudre ce nouveau problème, nous pouvons utiliser le code ci-dessous ;

**import** numpy **as** np

**from** scipy**.**optimize **import** minimize

**import** math

**def** objective**(**x**):**

**return** 15**\***x**[**0**]** **+** 20**\***x**[**1**]** **-** 1**/**2 **\*** math**.**sqrt**((**2**\***x**[**0**]+**3**\***x**[**1**]+**1**)\***math**.**pi**/**2**)** **-** 1**/**3 **\*** math**.**sqrt**((**x**[**0**]+**2**\***x**[**1**]+**1**)\***3**\***math**.**pi**/**4**)**

**def** constraints**(**x**):**

**return** **[**2**\***x**[**0**]** **+** 3**\***x**[**1**]** **-** 180**,** x**[**0**]** **+** 2**\***x**[**1**]** **-** 120**,** 2**\***x**[**0**]** **+** 3**\***x**[**1**]** **-** 80**]**

x0 **=** np**.**array**([**0**,** 0**])**

bnds **=** **[(**0**,** **None),** **(**0**,** **None)]**

con **=** **{**'type'**:** 'ineq'**,** 'fun'**:** constraints**}**

sol **=** minimize**(**objective**,** x0**,** method**=**'SLSQP'**,** bounds**=**bnds**,** constraints**=**con**)**

**print(**"Solution: x1 ="**,** sol**.**x**[**0**],** ", x2 ="**,** sol**.**x**[**1**])**

**print(**"Coût optimal : "**,** sol**.**fun**)**

La solution optimale donnée par ce programme est comme suit :

**Solution: x1 = 2.1437074337882223e-11 , x2 = 60.00000000003955**

**Coût optimal : 1185.940894499238**

**Partie 3 : Programmation non linéaire :**

Calculons le coût de réalisation C de l’autoroute : le coût entre A et P est de 4.105(10 − y) e tandis qu’il est de 5.105 x e entre P et C. Finalement, C(x, y) = 400000(10 − y) + 500000 x. Les contraintes s’écrivent x^2 = y^2 + 3^2 (théorème de Pythagore dans le triangle PBC), 3 ≤ y ≤ 10 compte tenu de la contrainte liée à la pollution de B et x ≥ 0. On notera que si les contraintes précédentes  
sont vérifiées, alors en particulier x ≤ √10^2 + 3^2, autrement dit x est inférieur à la distance AC.

