Algorithmes d’optimisation

Pr. Faouzia Benabbou (faouzia.benabbou@univh2c.ma)
Département de mathématiques et Informatique

Master Data Science & Big Data

2024-2025

FACULTE DES SCIENCES BEN M'SICK
UNIVERSITE HASSAN II DE CASABLANCA

Plan du Module: Algorithmes d’optimisation

01 Introduction

02 Rappels mathématiques

03 Algorithmes d’Optimisation Sans Contraintes

04 Algorithmes d’Optimisation avec Contraintes

05 Programmation Linéaire

06 Optimisation combinatoire et globale

07 Méthodes heuristiques et métaheuristiques

Les algorithmes d’optimisation

Classes des algorithmes

Optimisation

- Sans contraintes
  - Méthodes Analytiques
  - Méthodes numériques
    - Méthode sans gradient
    - Méthode de descente de gradient:
      - ✓ Méthode de gradient à pas fixe et optimal
      - ✓ Méthode de gradient conjugué
      - ✓ Méthode de Newton
      - ✓ Méthodes du gradient quasi-newton
- Sous contrainte
  - Méthodes générales de résolutions de PCE, PCI
  - Programmation linéaire
  - Optimisation combinatoire et globale
  - Méthodes heuristiques et métaheuristiques
  - Algorithmes multi-objectifs

Programmation Linéaire

- La programmation linéaire est un cas particulier de problèmes d'optimisation où la fonction **objective et les** contraintes **sont linéaire, et** les variables de décision sont **non-négatives**.
- Elle consiste, de manière générale, à résoudre des problèmes d’allocation optimale de ressources limitées entre des activités concurrentes.
- Elle trouve des applications concrètes dans des domaines tels que la logistique, la finance, la production ou encore la gestion des ressources.

Programmation Linéaire

- **Exemple classique**
  - Polly a besoin d'un **régime alimentaire équilibré**.
  - Chaque jour, elle doit consommer **au moins** une certaine quantité de nutriments : protéines, lipides, glucides.
  - Elle choisit 6 aliments qui semblent être des sources **économiques** de ces nutriments, rassemblés dans le tableau ci-dessous :

	Portion	Énergie (kcal)	Protéines (g)	Calcium (mg)	Prix/portion
Céréales	28g	110	4	2	3
Poulet	100g	205	32	12	24
Oeufs	2 gros	160	13	54	13
Lait entier	237cc	160	8	285	9
Tarte	170g	420	4	22	20
boeuf et haricots	260g	260	14	80	19

Programmation Linéaire et non Linéaire

- **Exemple classique.**
  - Chaque type apporte une certaine quantité de nutriments, et chaque portion a un **coût**.
  - Elle estime que ce serait trop difficile à supporter et décide de se limiter à :
    - ✓ **Flocons d'avoine** : maximum 4 portions par jour
    - ✓ **Poulet** : maximum 3 portions par jour
    - ✓ **Œufs** : maximum 2 portions par jour
    - ✓ **Lait** : maximum 8 portions par jour
    - ✓ **Tarte aux cerises** : maximum 2 portions par jour
    - ✓ **bœuf aux haricots** : maximum 2 portions par jour
  - Polly doit dépenser en nourriture pour obtenir toute l'énergie (2000 kcal), les protéines (55 g) et le calcium (800 mg) dont elle a besoin chaque jour.
  - Polly peut-elle trouver une solution qui va **minimiser les dépenses** tout en couvrant ses **besoins nutritionnels**?

Programmation Linéaire

- **Exemple classique. Modélisation du problème**

Soit x_1, x_2, x_3, x_4, x_5, et x_6 représentant : **Flocons, Poulet, Œufs, Lait, Tarte aux cerises, bœuf aux haricots.**

✓ Les premières contraintes sont sur les portions par jour qu'on peut formuler comme :

$$0 \leq x_1 \leq 4,\ 0 \leq x_2 \leq 3,\ 0 \leq x_3 \leq 2,\ 0 \leq x_4 \leq 8,\ 0 \leq x_5 \leq 2,\ 0 \leq x_6 \leq 2$$

✓ la deuxième contrainte porte sur l'énergie (2000 kcal), les protéines (55 g) et le calcium (800 mg) qui se traduisent par :

$$110\,x_1 + 205\,x_2 + 160\,x_3 + 160\,x_4 + 420\,x_5 + 260\,x_6 \geq 2000$$

$$4\,x_1 + 32\,x_2 + 13\,x_3 + 8\,x_4 + 4\,x_5 + 14\,x_6 \geq 55$$

$$2\,x_1 + 12\,x_2 + 54\,x_3 + 54\,x_4 + 285\,x_5 + 80\,x_6 \geq 800$$

Programmation Linéaire

- **Exemple classique. Modélisation du problème**

fonction à minimiser est la suivante :

$$3\ x_1 + 24\ x_2 + 13\ x_3 + 9\ x_4 + 20\ x_5 + 19\ x_6$$

- Rappel : Une fonction est linéaire si elle est de la forme : $\sum_{i=1}^{i=p} \boldsymbol{a_i x_i}$
- Dans ce cas de figure, on remarque que:
  - On a des contraintes d’inégalité
  - Les contraintes sont **linéaires**
  - La fonction objective est **linéaire**

Programmation Linéaire

- **Définition.** La forme générale d'un problème de programmation linéaire est la suivante:

$$\begin{cases} \max\limits_{x\in\mathbb{R}^n} f(\mathrm{x}) = \sum\limits_{j=1}^{j=n} c_j x_j \text{ , } f \textit{ une fonction objective.} \\ \sum\limits_{j=1}^{j=n} a_{ij}x_j \le b_i \ \ \mathbf{pour\ i = 1,..,p.} \\ \sum\limits_{j=1}^{j=n} a_{ij}x_j \ \ge \ b_i \ \ \mathbf{pour\ i = p+1,..,q.} \\ \sum\limits_{j=1}^{j=n} a_{ij}x_j = \ b_i \ \ \mathbf{pour\ i = q+1,..,m.} \\ x_j \ge 0 \ \mathbf{pour\ j = 1,..,n} \end{cases}$$

Programmation Linéaire

- Les programmes linéaires reposent sur les hypothèses implicites suivantes:
  - **Linéarité :** Les relations sont **linéaires** (fonction objective et contraintes).
  - **Proportionnalité**: la contribution des variables individuelles dans la fonction objective et les contraintes est proportionnelle à leur valeur.
  - **Additivité**: l'additivité signifie que la contribution totale de toutes les variables, tant dans la fonction objective que dans les contraintes, est égale à la somme des contributions individuelles de chaque variable; c.-à-d. il n'y a pas d'interaction entre les différentes variables de décision
  - **Divisibilité:** les variables de décision peuvent prendre n'importe quelle valeur numérique réelle (continues, binaires, entières) dans une plage spécifiée par les contraintes.
    - ✓ si les variables sont limitées à des valeurs entières, le problème est formulé et résolu avec programmation linéaire en nombres entiers – PLNE
  - **Certitude**: nous supposons que les valeurs des paramètres du modèle sont connues avec certitude.

Programmation Linéaire

- **Solutions d'un problème de PL**

Définitions.

- Une **solution** en PL est une attribution de valeurs numériques à chacune des variables de décision du modèle.
- Une solution **réalisable** (Feasible solution) est une solution qui satisfait à toutes les contraintes du modèle, y compris les bornes sur les variables.
- **L'ensemble ou la région réalisable** (Feasible set) est l'ensemble de toutes les solutions réalisables. En PL, cet ensemble est souvent un **polyèdre convexe**.
- Une **solution optimale** (Optimal solution) est la solution réalisable qui maximise ou minimise la fonction objective, selon le but du problème.
- La solution d'un PL dépend :
  - ✓ de la région des solutions réalisables (vide, bornée ou non bornée)
  - ✓ le type d'optimisation (maximisation ou minimisation),
- La solution optimale du programme linéaire est soit **unique**, **multiple**, **infinie** ou **pas de solution.**

Programmation Linéaire et non Linéaire

- **Solutions d'un problème de PL**

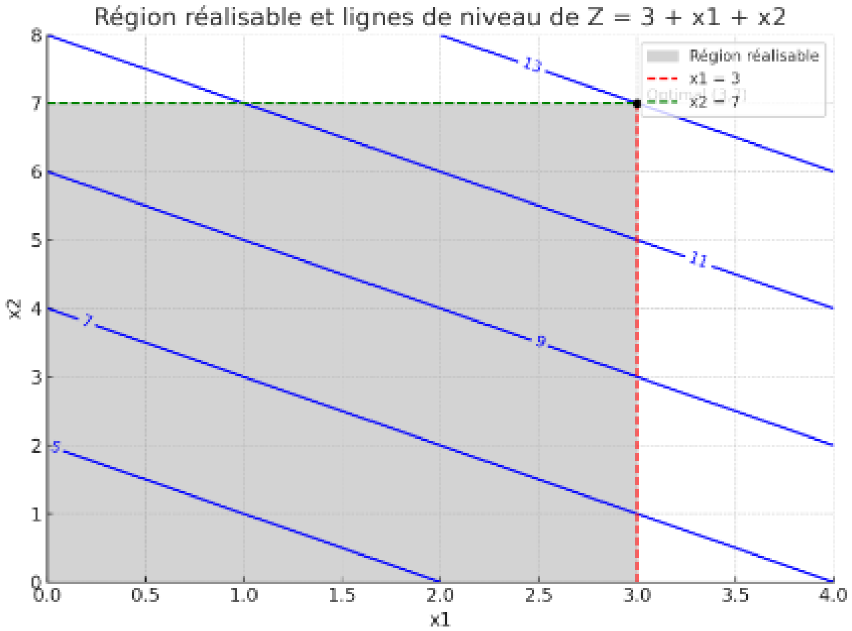
Exemple.

$$\text{Problème 1 :}\begin{cases} \max \boldsymbol{z} = \mathbf{3} + \mathbf{x1} + \mathbf{x2} \\ \mathbf{x1} \leq \mathbf{3} \\ \mathbf{x2} \leq \mathbf{7} \\ \mathbf{x1}, \mathbf{x2} \geq \mathbf{0} \end{cases}$$

Programmation Linéaire

- **Solutions d'un problème de PL**

$$\text{Problème 1 :} \begin{cases} 3 + x1 + x2 \\ x1 \leq 3 \\ x2 \leq 7 \\ x1, x2 \geq 0 \end{cases} \quad \text{trouver}$$



✓ **Solution admissible ?** Oui (x1=0,x2)

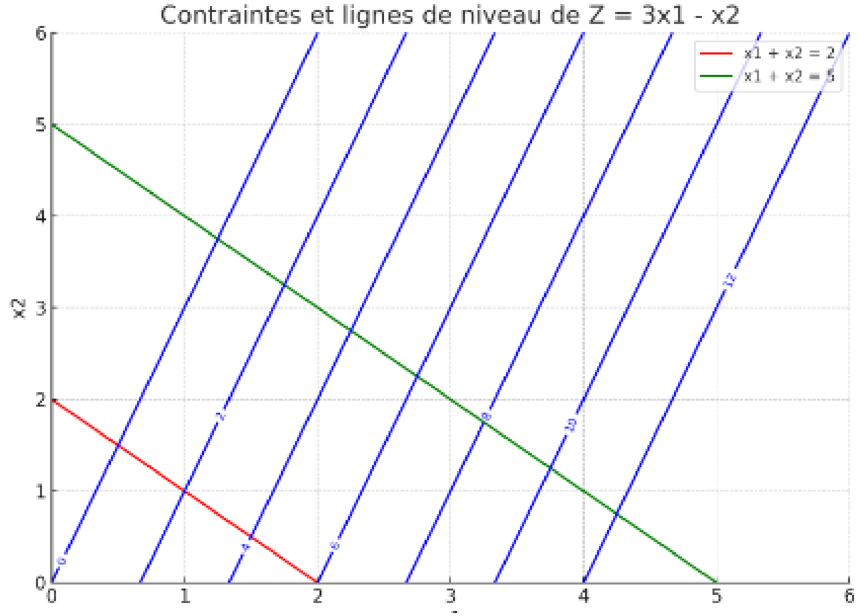
✓ **Solution optimale ?** Oui (x1=3,x2=7)

Programmation Linéaire

- **Solutions d'un problème de PL**

$$\text{Problème 2 :}\begin{cases} \max z = 3x1 - x2 \\ x1 + x2 \leq 2 \\ -2x1 - 2x2 \leq -10 \\ x1, x2 \geq 0 \end{cases}$$

trouver le max



- ✓ **Solution admissible ?** Non
- ✓ **Solution optimale ?** Non

Ici, on a : $x1+x2\leq 2$ et $x1+x2\geq 5$

⇒ Aucune solution ne satisfait les deux

Ce problème est non faisable (pas de solution admissible).

Programmation Linéaire

- **Solutions d'un problème de PL**

Problème 3 : $\begin{cases} Max\ z = x1\ - \text{x2} \\ -2x1 + x2 \leq -1 \\ -x1 - 2x2 \leq -2 \\ x1, x2 \geq 0 \end{cases}$

Analysons la **première contrainte** :

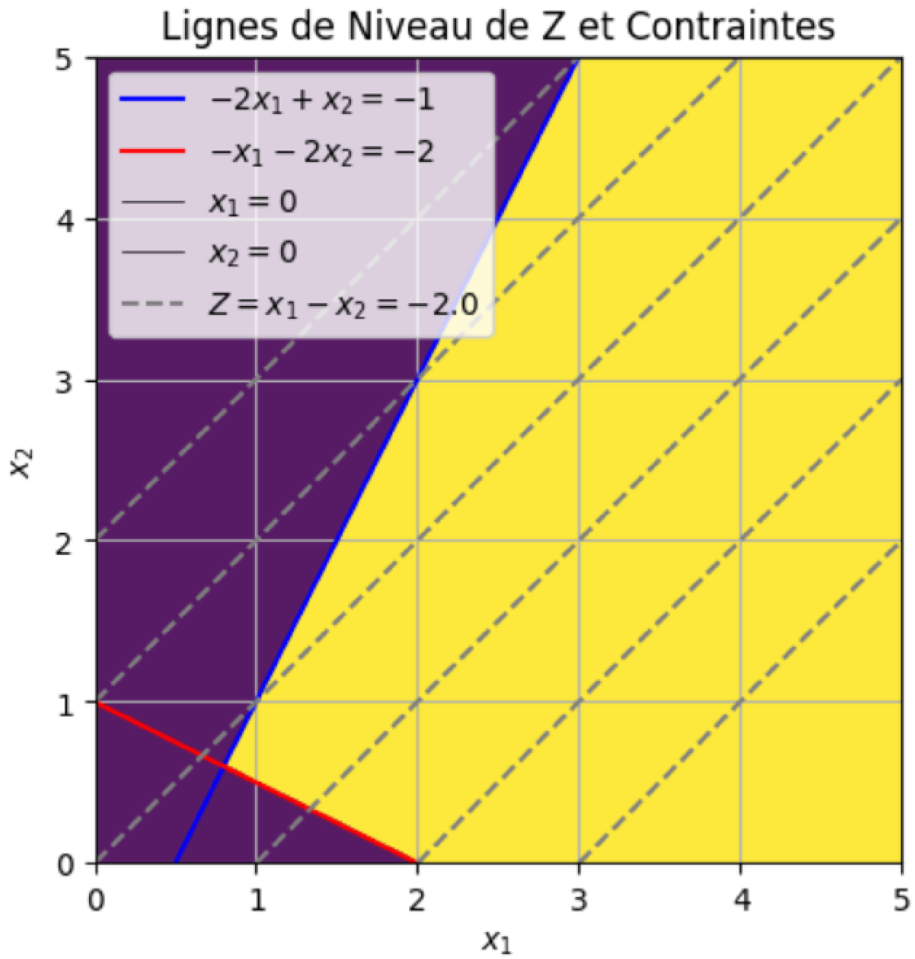
−2x1+x2≤−1⇒ x2≤2x1−1

−x1−2x2≤−2 ⇒ x1+2x2≥2

Testons d'autres points admissibles :

Pour x1=1, on a x2=1

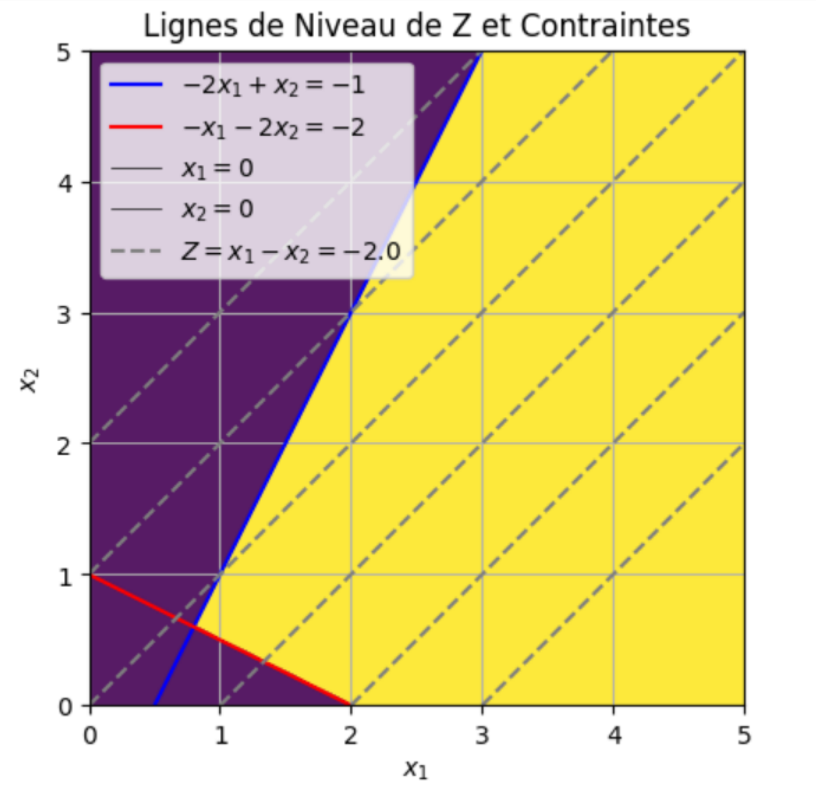
- −2(1)+1=−1 → OK
- −1−2(1)=−3≤−2 → OK
- x1,x2≥0→ OK



Programmation Linéaire

- **Solutions d'un problème de PL**

$$\text{Problème 3 :} \begin{cases} x1 \ - \mathrm{x2} \\ -2x1 + x2 \leq -1 \\ -x1 - 2x2 \leq -2 \\ x1, x2 \geq 0 \end{cases} \text{trou}$$



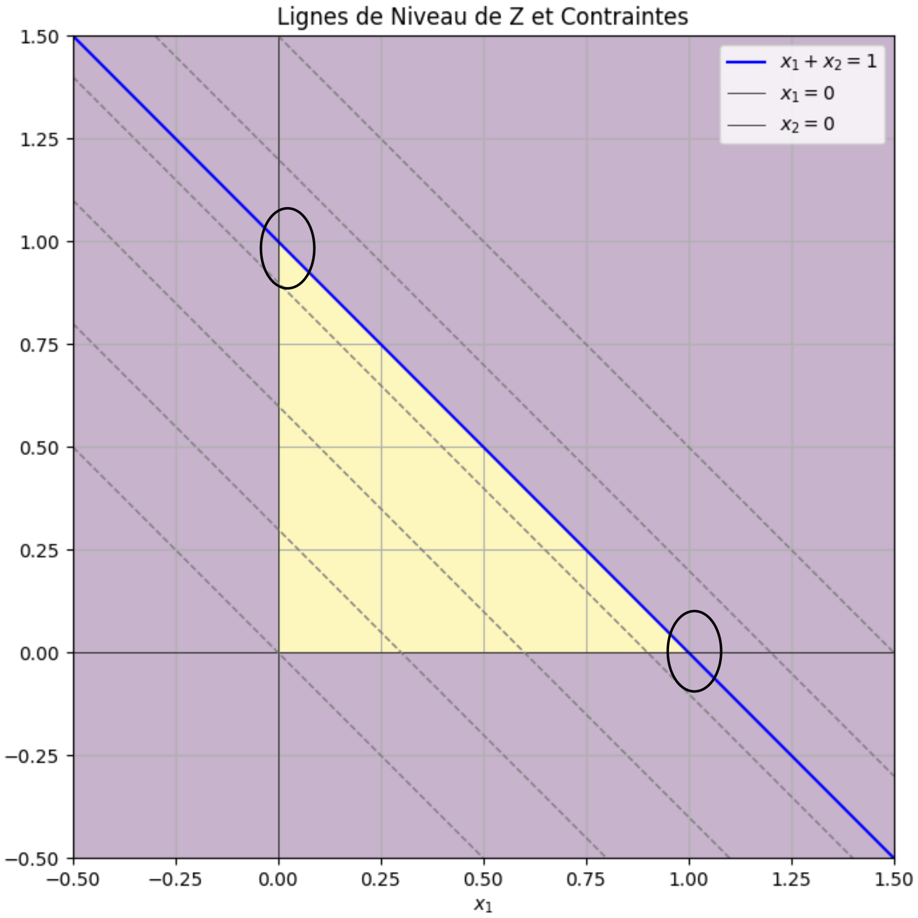
✓ **Solution admissible ?** oui
✓ **Solution optimale ?** No

Donc une solution admissible existe mais le problème est non borné : il existe une infinité de solutions faisables qui améliorent indéfiniment la valeur de la fonction objectif, sans jamais atteindre un maximum global.

Programmation Linéaire et non Linéaire

- **Solutions d'un problème de PL**

$$\text{Problème 4 :}\begin{cases} x1 + \mathrm{x2} \\ x1 + x2 \leq 1 \\ x1, x2 \geq 0 \end{cases}$$



✓ **Solution admissible ?** Oui (1, 0)
✓ **Solution optimale ?** Oui mais pas unique

Les points (1,0), (0,1) des s**olutions optimales.**

Programmation Linéaire

- **Points extrêmes (sommets) de la région admissible**

Définition d'un point extrême.

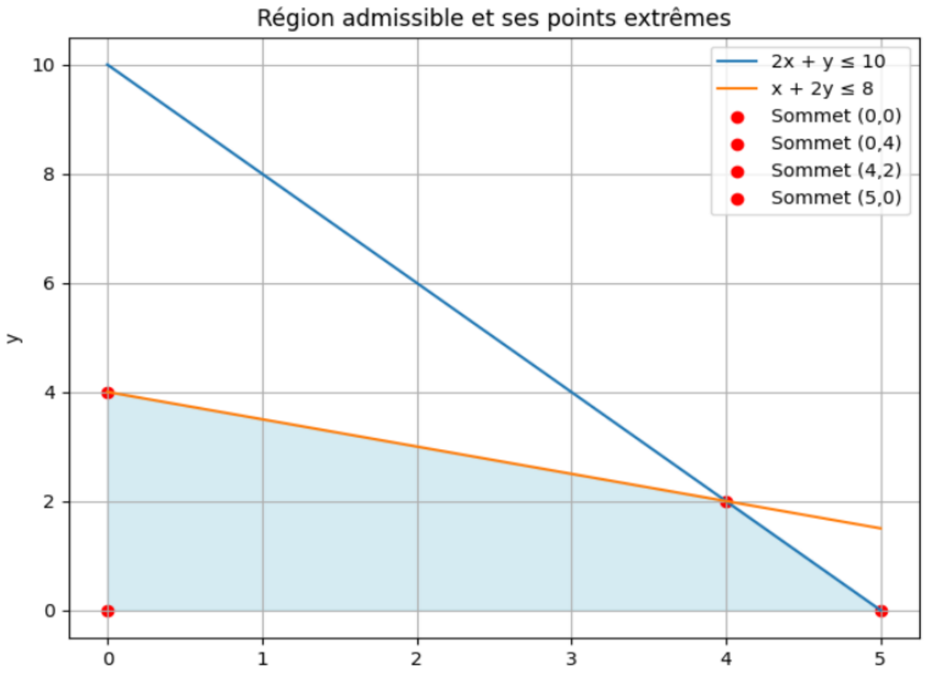
- Un point extrême (ou sommet) est une solution admissible qui ne peut pas être exprimée comme une combinaison convexe d'autres points de la région admissible.
- Autrement, il n'existe pas deux points distincts $y \in S$ et $z \in S$ (où $y \neq z$) et un scalaire $\lambda \in (0,1)$ tels que $x=\lambda y+(1-\lambda)z$.
- En 2D, ce sont les coins du polygone formé par les contraintes.
- En 3D, ce sont les intersections des hyperplans définis par les contraintes.
- La région admissible a un nombre fini de points extrêmes qui sont candidats naturels pour être des solutions optimales.

Programmation Linéaire et non Linéaire

- **Points extrêmes (sommets) de la région admissible**

Exemple.

$$\begin{cases} \max\limits_{x}\ x1 + x2 \\ 2x1 + x2 \leq 10 \\ x1 + 2x2 \leq 8 \\ x1, x2 \geq 0 \end{cases}$$



- La figure suivante représente la région admissible sous forme de polygone.
- Les somment de la région sont : (0,0), (0,4), (4,2), et (5,0). Si on maximise/minimise une fonction linéaire, l'optimum se trouvera sur l'un d'eux.

Programmation Linéaire

- **Théorème fondamental de la PL.** Si un problème de programmation linéaire admet une solution optimale, alors il existe au moins une solution optimale qui se trouve en un point extrême de la région admissible (correspondant à un sommet du polyèdre).
- Ce que cela signifie qu'il n'est **pas nécessaire** de chercher toutes les solutions admissibles, on peut limiter la recherche d'une solution optimale aux points extrêmes (au lieu de toute la région).
- C'est la base de l'algorithme du **Simplexe**, qui parcourt ces sommets pour trouver l'optimum.
- Si plusieurs sommets donnent la même valeur optimale, il y a multiplicité de solutions optimales.
- Si la région admissible est non bornée, la fonction objective peut diverger vers $+\propto$ dans le cas d'une maximisation, dans ce cas on aura pas de solution optimale.

Programmation Linéaire

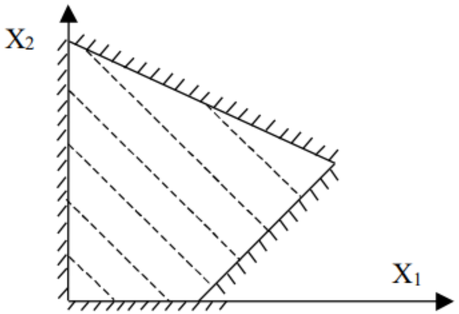
- **Méthode graphique : Géométrie de la programmation linéaire**
  - Il existe diverses techniques de résolution des PL parmi lesquelles la méthode **graphique** ou **géométrique** est la plus rapide et la plus simple mais aussi la plus limitée, car dès lors que le nombre de variables de décision ou de contraintes dépasse 2, elle devient impraticable.
  - C’est pourquoi divers chercheurs se sont efforcés de mettre au point une méthode de calcul algorithmique comme la méthode du simplexe qui est efficace quel que soit le nombre des variables et des contraintes.

Programmation Linéaire

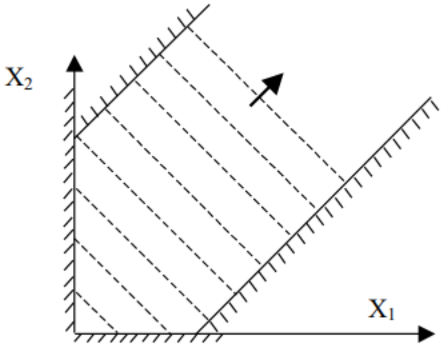
- **Méthode graphique : Géométrie de la programmation linéaire**
  - Généralement il y a quatre étapes à suivre pour résoudre un programme linéaire en utilisant la méthode graphique:
    - ✓ Représenter graphiquement les droites (équations provenant des inéquations) c.-à-d. Tracer les contraintes (fonctionnelles et de non-négativité) et déterminer le demi-plan fermé satisfaisant chaque contrainte.
    - ✓ Déterminer le **polygone de contraintes** comme l'intersection de toutes ces régions, en tenant compte des contraintes de non-négativité.
    - ✓ Tracer les niveaux du vecteur de coûts $\boldsymbol{C}$ de la fonction objective.
    - ✓ Remplacer successivement les coordonnées de chaque point de la région dans la fonction objective afin de rechercher la (les) solution(s) optimale(s) (si elle existe).

Programmation Linéaire

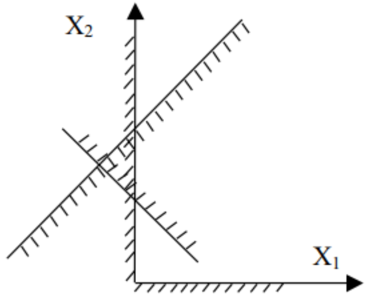
- **Méthode graphique : Géométrie de la programmation linéaire**
  - 3 situations possibles peuvent résulter, le polygone de contraintes peut être:



Borné : Il s'agit d'une région fermée avec un nombre fini de sommets (coins), par conséquent le Pl admet une solution optimale (qui est un sommet du polyèdre) non nécessairement unique.



Non borné : Il s'étend à l'infini dans au moins une direction, le domaine est non vide mais la fonction objective n'est pas majorée sur ce domaine et le max de f = +∞.



Vide : Le domaine est vide = ø, si les contraintes sont contradictoires et qu'aucun point ne peut les satisfaire toutes en même temps. Le PL n'a pas de solutions.

Programmation Linéaire

- **Méthode graphique : Géométrie de la programmation linéaire**

Exemples.

1. Soit le PL suivant, qu'on interprétera comme le calcul des quantités à produire x1 et x2 de deux produits pour maximiser un profit z.

$$\begin{cases} \text{Max } z = x_1 + 2x_2 \\ x_1 + x_2 \leq 6 \\ x_2 \leq 3 \\ x_1, x_2 \geq 0 \end{cases}$$

Programmation Linéaire

- **Méthode graphique : Géométrie de la programmation linéaire**
- **Exemples.**

$$\begin{cases} \text{Max z} = x_1 + 2x_2 \\ x_1 + x_2 \leq 6 \\ x_2 \leq 3 \\ x_1, x_2 \geq 0 \end{cases}$$

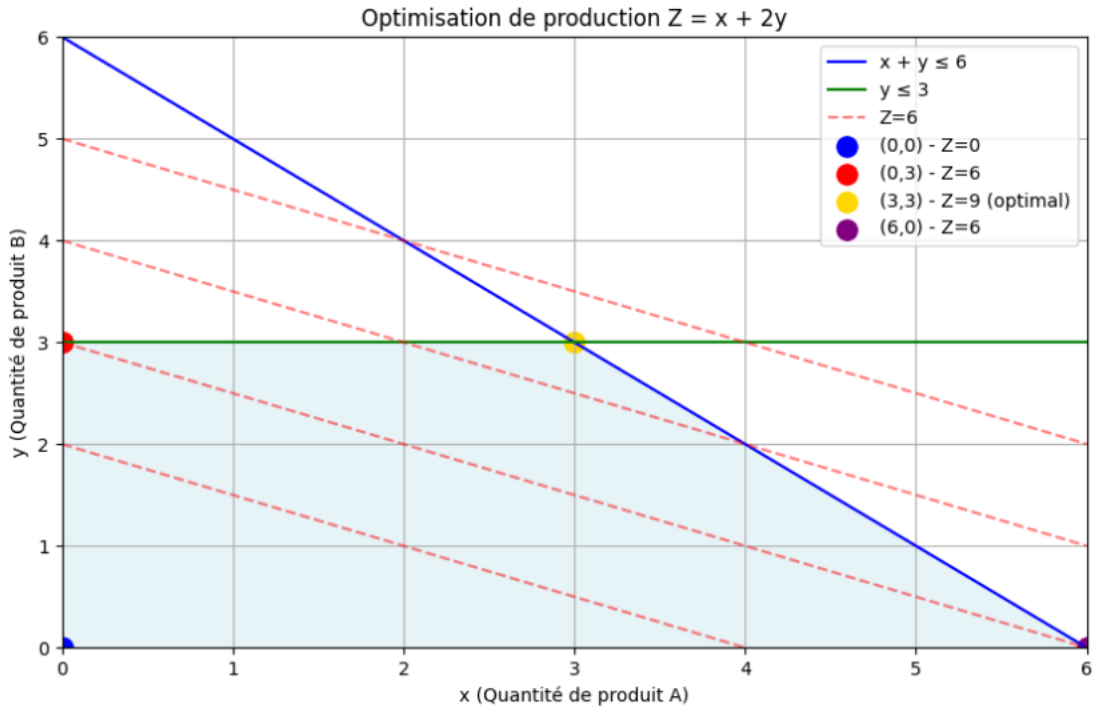
  - On délimite le domaine des solutions réalisables, qui forme un polygone convexe, en traçant les droites d'équations :
  - On trace les droites relatives aux contraintes:
    - ✓ D1: xl + x2 = 6 (0, 6) et (6, 0)
    - ✓ D2: x2 = 3
    - ✓ D3 : $x_1 \geq 0$
  - On tracer certaines lignes de niveaux arbitraire de f (4, 6, 8, 10) .

Programmation Linéaire

- **Méthode graphique : Géométrie de la programmation linéaire**
- **Exemples.**

$$\begin{cases} \text{Max z} = x_1 + 2x_2 \\ x_1 + x_2 \leq 6 \\ x_2 \leq 3 \\ x_1, x_2 \geq 0 \end{cases}$$

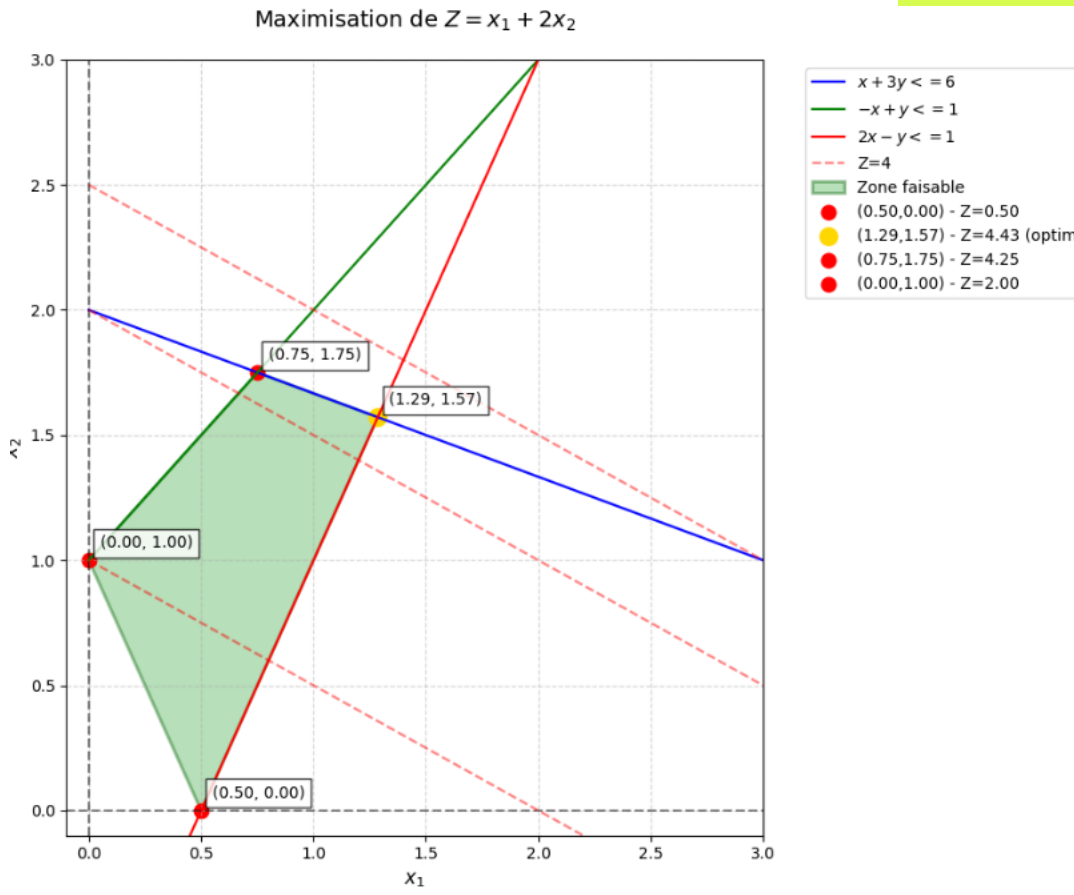
Le Polyèdre a 4 sommets : (0,0), (0,3), (3,3), (6,0) et le sommet 3 présente la production maximale.



Programmation Linéaire

- **Méthode graphique : Géométrie de la programmation linéaire**

- **Exemples.** $\begin{cases} \text{Max z } = x_1 + 2x_2 \\ x_1 + 3x_2 \leq 6 \\ -x_1 + x_2 \leq 1 \\ 2x_1 - x_2 \leq 1 \\ x_1, x_2 \geq 0 \end{cases}$



- On trace les droites relatives aux contraintes:
  - D1: x1 + 3x2 = 6 (0, 2) et (6, 0)
  - D2: -x1 + x2 = 1 (0, 1) et (1, 0)
  - D3: 2x1 – x2 = 4 (0, 4) et (2, 0)
  - D4 : $x_1 \geq 0$
  - D5 : $x_2 \geq 0$

Parmi les Sommets admissibles et valeurs de Z:
Le Sommet 4: (0.00, 1.00) - Z = 2.00
est le point optimal.

Programmation Linéaire

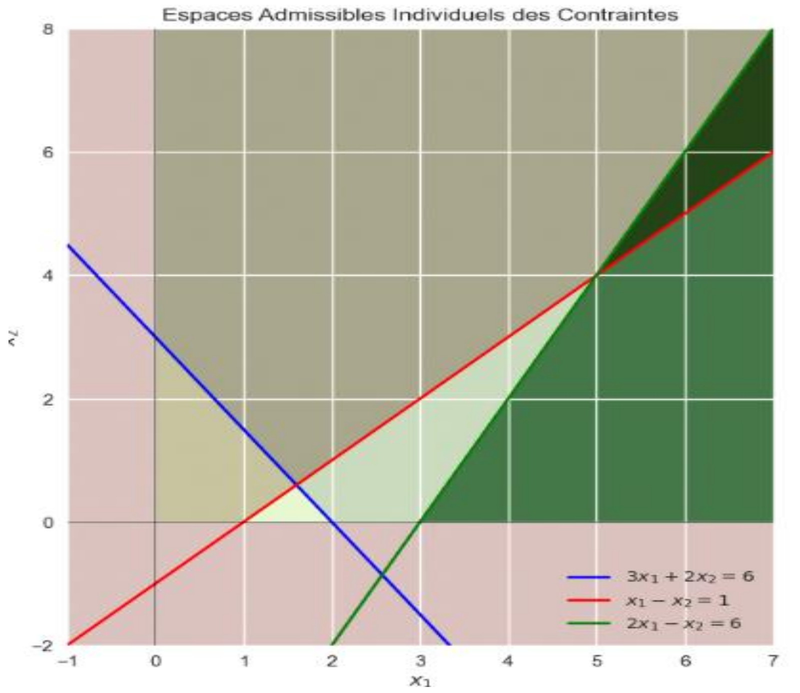
- **Méthode graphique : Géométrie de la programmation linéaire**

Exemples.

$$\begin{cases} \text{Max z} = x_1 + 2x_2 \\ 3x_1 + 2x_2 \leq 6 \\ x_1 - x_2 \leq 1 \\ 2x_1 - x_2 \geq 6 \\ x_1, x_2 \geq 0 \end{cases}$$

La contrainte $2x_1 - x_2 \geq 6$ implique que $x_1 \geq 3 + \frac{x_2}{2} \geq 3$, doit être au moins 3 (en plus ue si $x_1 < 3$ et $x_2 \geq 0$, $2x_1 - x2$ sera toujours inférieur à 6).

Considérons maintenant la première contrainte $3x_1 + 2x_2 \leq 6$. Si $x_1 > 3$ et $x_2 \geq 0$, alors $3x_1 + 2x_2 \geq 3(3)+2(0)=9$, ce qui viole l'inégalité $3x_1 + 2x_2 \leq 6$.



La région admissible définie par ce système d'inégalités est **vide, le problème principal est entre D1 et D3.**

Programmation Linéaire

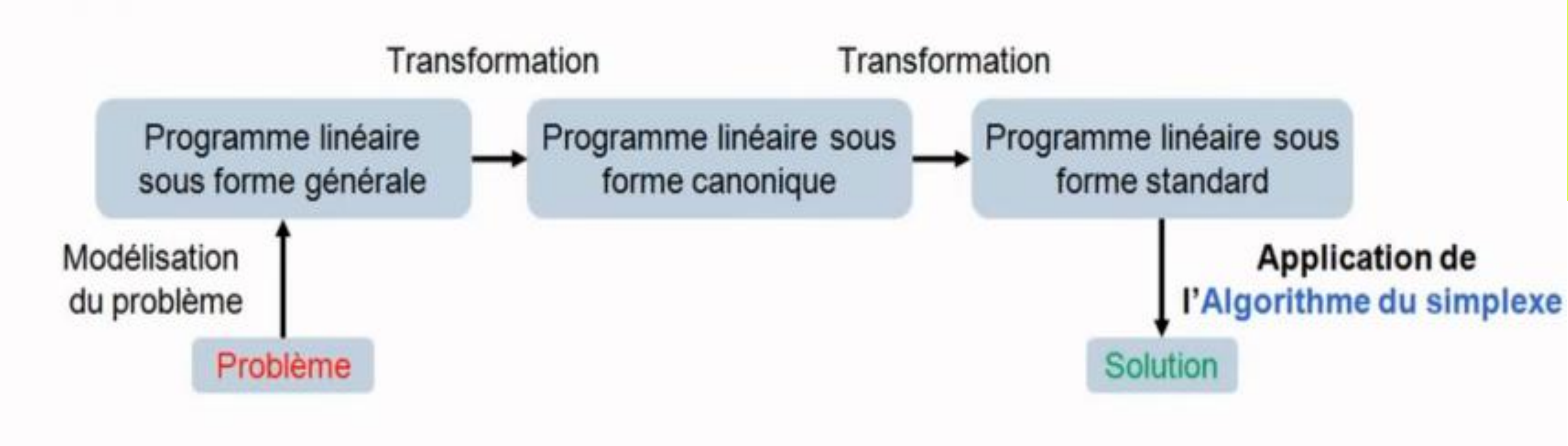
- **Méthode du Simplexe**
  - L'**algorithme du simplexe** est un algorithme fondamental en programmation linéaire (PL) pour résoudre les problèmes d'optimisation linéaire.
  - Développé par George Dantzig à partir de 1947, il reste l'une des méthodes les plus utilisées pour trouver une solution optimale à un PL, en particulier pour les problèmes de taille modérée.

Programmation Linéaire

- **Méthode du Simplexe**
  - **Concept Fondamental : Exploration des Sommets**
    - ✓ L'algorithme du simplexe exploite la propriété fondamentale de la programmation linéaire : si une solution optimale existe, elle se trouve toujours à un **sommet** (ou point extrême) du **polytope des contraintes** (la région réalisable définie par les inégalités et les contraintes d'égalité).
    - ✓ l'algorithme du simplexe se déplace itérativement d'un sommet admissible à un sommet admissible voisin, en améliorant la valeur de la fonction objective à chaque étape.
    - ✓ Il s'arrête lorsqu'il atteint un sommet où aucune amélioration n'est possible, ce qui correspond à la solution optimale.

Programmation Linéaire

- ## Méthode du Simplexe
  - **Mise sous Forme Standard:** Avant d'appliquer l'algorithme du simplexe, le problème de PL doit être converti en **forme standard**.



Programmation Linéaire

- **Méthode du Simplexe**
  - **Mise sous Forme Standard**
    - ✓ La fonction objective est exprimée sous forme de maximisation (si le problème original est une minimisation, on maximise l'opposé de la fonction objective -f).
    - ✓ Les constantes à droite des contraintes (b) doivent être positives ou nulles. Si b < 0, multiplier la contrainte entière par -1 (et inverser le sens de l'inégalité si nécessaire).
    - ✓ Transformation des contraintes **d'inégalité en égalités** en introduisant des variables **d'écart** (pour les contraintes ≤) ou des variables **d'excédent** (pour les contraintes ≥).
      - Pour les contraintes de type $\sum_{j=1}^{j=n} a_{ij}x_j \leq b_i$, on ajoute une variable d'écart $s_i \geq 0$: $\sum_{j=1}^{j=n} a_{ij}x_j + s_i = b_i$.
      - Pour $\sum_{j=1}^{j=n} a_{ij}x_j \geq b_i$, on soustrait une variable d'excédent $e_i \geq 0$: $\sum_{j=1}^{j=n} a_{ij}x_j - e_i = b_i$.
    - ✓ Toutes les variables doivent être non négatives ($x_j \geq 0$, $s_i \geq 0$, $e_i \geq 0$).
      - Toute variable x_j négative ou nulle, i.e. $x_j \leq 0$, sera remplacée par une variable x_j^+ tel que $x_j^+ = -x_j$;
      - Toute variable x_j qui n'est soumise à aucune condition de signe ($x_j \in \mathbb{R}$) sera remplacée par deux variables non négative x_j^+ et x_j^- tel que $x_j = x_j^+ - x_j^-$ avec où $x_j^+ = \max[0, x_j]$, $x_j^- = \max[0, -x_j]$. (tout réel est différence de deux réels positifs ou nuls).

Programmation Linéaire

- ## Méthode du Simplexe
  - ### Mise sous Forme Standard

Exemple 1.

Soit le problème de programmation linéaire suivant :

$$\begin{cases} \text{Max z} = 6x_1 - 3x_2 + x_3 \\ 4x_1 + 2x_2 + x_3 \leq 65 \\ x_1 + x_2 - x_3 \geq 5 \\ x_1 + x_2 = 10 \\ x_1 \geq 0, x_2 \leq 0 \end{cases}$$

1. Pour transformer dans le format canonique, les variables doivent être positives ou nulles :

$x_2 \leq 0$ et x3 libre ⇒ nous allons poser :

$y_1 = x_1$, $y_2 = -x_2$, $x_3 = y_3 - y_4$, où les variables y_i i = 1,2,3,4 sont toutes positives ou nulles.

2. Toutes les contraintes doivent être du type (=)

4y1 - 2y2 + y3 - y4 ≤ 65 devient : $4 y_1 - 2 y_2 + y_3 - y_4 + s_1 = 65$

$y_1 + y_2 + y_4 - y_3 \geq 5 \Rightarrow y_1 + y_2 + y_4 - y_3 - e_1 = 5$

Programmation Linéaire

- **Méthode du Simplexe**
  - **Mise sous Forme Standard**

Exemple 1.

La forme standard du problème est la suivante :

$$\begin{cases} \text{Min } z = 6y_1 - 3y_2 + y3 - y4 \\ 4\, y_1 - 2\, y_2 + y_3 - y_4 + s_1 = 65 \\ y_1 + y_2 + y_4 - y_3 - e_1 = 5 \\ y_1 + y_2 = 10 \\ y_1 \geq 0, y_2 \geq 0, y_3 \geq 0, \; y_4 \geq 0, s_1 \geq, e_1 \geq 0. \end{cases}$$

Programmation Linéaire

- **Méthode du Simplexe**
  - **Mise sous Forme Standard**

Exemple 1.

Soit le problème de programmation linéaire suivant :

$$\begin{cases} \text{Min } Z = 6\,x_1 + 14x_2 + 13\,x_3 \\ \frac{1}{2}x_1 + 2\,x_2 + x_3 \leq 24 \\ x_1 + 2x_2 + 4x_3 \leq 60 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{cases} \Rightarrow \begin{cases} \text{Min } Z = 6\,x_1 + 14x_2 + 13\,x_3 \\ \frac{1}{2}x_1 + 2\,x_2 + x_3 + s_1 = 24 \\ x_1 + 2x_2 + 4x_3 + s_2 = 60 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \\ s_1, s_2 \geq 0 \end{cases}$$

Programmation Linéaire

- **Méthode du Simplexe**
  - **Le Tableau du Simplexe**
    - ✓ L'algorithme du simplexe est généralement implémenté en utilisant un **tableau**.
    - ✓ Le tableau initial est construit à partir de la forme standard du problème.
    - ✓ Le nombre de **ligne** correspond au **nombre de contrainte** du problème plus la ligne de la fonction a maximiser.
    - ✓ Le nombre de **colonnes** correspond **au nombre total** des **variables**.

Programmation Linéaire

▪ Méthode du Simplexe

- **Le Tableau du Simplexe.** Le tableau initial contient les informations suivantes :
  - ✓ **Ligne de la fonction objective (ligne Z) :** Contient les coefficients de la fonction objective (**-Z** pour maximiser) et un 0 à la dernière colonne (valeur initiale de Z).
  - ✓ **Lignes de contraintes :** Chaque ligne représente une **contrainte d'égalité** et contient les **coefficients des variables**, la variable de base correspondante et la valeur du second membre bi (ex. $\sum_{j=1}^{j=n} a_{ij}x_j + s_i = \boldsymbol{b_i}$).
  - ✓ **Variables de base :** Les variables qui forment une matrice identité dans les colonnes correspondantes du tableau simplexe. Initialement, ce sont **les variables d'écart ou d'éxcédent** et vont changer au court de l'exécution de l'algorithme de simplexe.
  - ✓ **Variables hors base :** Ce sont toutes les autres variables du problème qui ne sont pas des variables de base (à l'initialisation sont les **variables initiales** du problème).

Programmation Linéaire

- **Méthode du Simplexe**
  - **Le Tableau du Simplexe.**

Exemple. $\begin{cases} \text{Max z} = 30x_1 + 50x_2 \\ 3x_1 + 2x_2 \leq 1800 \\ x_1 \leq 400 \\ x_2 \leq 600 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$ $\Longrightarrow$ Format standard

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 1800 \\ x_1 + x_4 = 400 \\ x_2 + x_5 = 600 \\ -\text{z} = -30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Programmation Linéaire

- **Méthode du Simplexe**
  - **Le Tableau du Simplexe.**

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 1800 \\ x_1 + x_4 = 400 \\ x_2 + x_5 = 600 \\ \text{z} - 30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Solution admissible (0,0)

Variables hors base (x_1, x_2) — Variables d'écart (x_3, x_4, x_5)

E_{ij} — Colonne j, Ligne i

	x_1	x_2	x_3	x_4	x_5	b
x_3	3	2	1	0	0	1800
x_4	1	0	0	1	0	400
x_5	0	1	0	0	1	600
-Z	-30	-50	0	0	0	0

Variables de base initiaux (x_3, x_4, x_5)

Coefficient de - Z (ligne -Z)

Programmation Linéaire

Méthode du Simplexe

L'algorithme 14. Simplexe cas de maximisation

1. **Initialisation :** Mettre le problème sous forme standard

2. **Construire le tableau du simplexe**

   a) **si état initial** identifier une solution de base réalisable initiale

   b) dans le cas maximisation prendre les coefficients de -Z.

3. **Test d'Optimalité :**
   a) Si tous les coefficients dans la ligne - z sont $\geq 0 \rightarrow$ solution optimale atteinte
   b) Sinon, passer à l'étape

4. **Choix de la Variable Entrante :** Sélectionner une variable hors base avec le coefficient le plus négative dans la ligne -Z.

5. **Choix de la Variable sortante :**
   a) Pour chaque contrainte i où le coefficient de la colonne entrante $(E_i c)_i$ est > 0, calculer: $\theta_i = b_i / E_i c$
   b) Choisir la ligne avec le plus petit θ_i positif

6. **Pivotage :**
   a) **Normalisation** : Diviser la ligne pivot par l'élément pivot pour obtenir 1 ($New\_E_{lj}=E_{lj}/E_{lc}$) c'est la nouvelle ligne pivot; (le pivot E_{lc} est l'intersection de la ligne et colonne pivot).
   b) **Élimination**: Calculer les nouvelles valeurs du tableau : Pour toutes les autres lignes (y compris z): Nouvelle ligne i = Ligne actuelle i - (coefficient de la colonne pivot de la ligne i × nouvelle Ligne pivot) $(New\_E_{ij}= E_{ij} - E_{ic} * (New\_E_{lj})_j)$

7. **Itération :** Retourner à l'étape 3 avec le nouveau tableau du simplexe jusqu'à satisfaction du critère d'optimalité.

Programmation Linéaire

- **Méthode du Simplexe**

L'algorithme 14. Simplexe cas de minimisation

1. **Initialisation :** Mettre le problème sous forme standard

2. **Construire le tableau du simplexe**

   a) si état initial identifier une solution de base réalisable initiale

   b) dans le cas minimisation prendre les coefficients de Z.

3. **Test d'Optimalité :**
   a) Si tous les coefficients dans la ligne z sont $\leq 0 \rightarrow$ solution optimale atteinte
   b) Sinon, passer à l'étape

4. **Choix de la Variable Entrante :** Sélectionner une variable hors base avec le coefficient le plus positif dans la ligne Z.

5. **Choix de la Variable sortante :**
   a) Pour chaque contrainte i où le coefficient de la colonne entrante $(E_i c)_i$ est > 0, calculer: $\theta_i = b_i / E_i c$
   b) Choisir la ligne avec le plus petit θ_i positif

6. **Pivotage :**
   a) Normalisation : Diviser la ligne pivot par l'élément pivot pour obtenir 1 ($New\_E_{lj}=E_{lj}/E_{lc}$) c'est la nouvelle ligne pivot; (le pivot E_{lc} est l'intersection de la ligne et colonne pivot).
   b) **Élimination**: Calculer les nouvelles valeurs du tableau : Pour toutes les autres lignes (y compris z): Nouvelle ligne i = Ligne actuelle i - (coefficient de la colonne pivot de la ligne i × nouvelle Ligne pivot) ($New\_E_{ij}= E_{ij} - E_{ic} * (New\_E_{lj})_j$)

7. **Itération :** Retourner à l'étape 3 avec le nouveau tableau du simplexe jusqu'à satisfaction du critère d'optimalité.

Programmation Linéaire

- **Méthode du Simplexe**
  - **Choix de la Variable Entrante**
    - ✓ Sélectionner une variable **hors base** avec un coefficient positif dans la ligne Z.
    - ✓ Choisir la variable entrante avec Z **le plus petit**
    - ✓ S'il y a deux ou plus variables qui satisfont aux conditions précédentes choisir celle avec l'indice le plus petit (règle de bland).
    - ✓ Cette colonne sera la **colonne pivot**

Programmation Linéaire

- **Méthode du Simplexe**
  - **Choix de la Variable Sortante** :
    - ✓ Pour déterminer jusqu'à quelle valeur la variable entrante peut augmenter sans violer les contraintes, on effectue le test du ratio minimum.
    - ✓ Calculer le rapport $\mathbf{b_i / a_ij}$ pour chaque contrainte (ligne) où $a_ij > 0$
    - ✓ La variable de base correspondant à la ligne avec le **ratio minimum positif** est choisie comme variable sortante.
    - ✓ Cette variable quittera la base et deviendra hors base.
    - ✓ Sa ligne sera la **ligne pivot**
  - L'élimination vise à rendre tous les autres éléments de la colonne pivot égaux à 0

Programmation Linéaire

▪ Méthode du Simplexe

- **Le Tableau du Simplexe.**

Itération 1

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 1800 \\ x_1 + x_4 = 400 \\ x_2 + x_5 = 600 \\ z - 30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Solution admissible (0,0)

Étape 3

Colonne j

E_{ij}

Ligne i

Variables de base initiaux

Coefficient de - Z

	x_1	x_2	x_3	x_4	x_5	b
x_3	3	2	1	0	0	1800
x_4	1	0	0	1	0	400
x_5	0	1	0	0	1	600
-Z	-30	-50	0	0	0	0
	≤0	≤0				

Programmation Linéaire

▪ Méthode du Simplexe

- **Le Tableau du Simplexe.**

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 1800 \\ x_1 + x_4 = 400 \\ x_2 + x_5 = 600 \\ z - 30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Solution admissible (0,0)

Colonne j

E_{ij}

Ligne i

Étape 4: variable entrante

	x_1	x_2	x_3	x_4	x_5	b
x_3	3	2	1	0	0	1800
x_4	1	0	0	1	0	400
x_5	0	1	0	0	1	600
Z	-30	-50	0	0	0	0

Variables de base initiaux

Coefficient de - Z

La colonne avec la plus petite valeur

Programmation Linéaire

- ## Méthode du Simplexe
  - **Le Tableau du Simplexe.**

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 1800 \\ x_1 + x_4 = 400 \\ x_2 + x_5 = 600 \\ z - 30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Solution admissible (0,0)

Étape 5: variable sortante

Colonne j

E_{ij}

Ligne i

	x_1	x_2	x_3	x_4	x_5	b	Ratio bi/Eic
x_3	3	**2**	1	0	0	1800	900
x_4	1	**0**	0	1	0	400	+oo
x_5	0	**1**	0	0	1	600	600
-Z	-30	**-50**	0	0	0	0	0

le plus petite ratio

x_5 Variable sortante, ligne pivot

Valeur du pivot= Elc

La variable sortante x5 qui sera remplacée par x2.

Programmation Linéaire

- **Méthode du Simplexe**
  - **Le Tableau du Simplexe.**

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 1800 \\ x_1 + x_4 = 400 \\ x_2 + x_5 = 600 \\ z - 30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Solution admissible (0,0)

Etape 6: Pivotage, **Normalisation**

(New_E_{lj}=E_{lj}/E_{lc})

$$\begin{cases} E_{l1} = \frac{0}{1} = 0 \\ E_{l2} = \frac{1}{1} = 1 \\ E_{l3} = \frac{0}{1} = 0 \\ E_{l4} = \frac{0}{1} = 0 \\ E_{l5} = \frac{1}{1} = 1 \end{cases}$$

Colonne j

E_{ij}

Ligne i

	x_1	x_2	x_3	x_4	x_5	b
x_3	3	**2**	1	0	0	1800
x_4	1	**0**	0	1	0	400
x_2	**0**	**1**	**0**	**0**	**1**	**600**
-Z	-30	-50	0	0	0	0

Programmation Linéaire

- ## Méthode du Simplexe
  - **Le Tableau du Simplexe.**

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 1800 \\ x_1 + x_4 = 400 \\ x_2 + x_5 = 600 \\ z - 30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Solution admissible (0,0)

Etape 6: Pivotage, **élimination**

Nous devons calculer les autres valeurs du tableau pour que les autres valeurs de la colonne du pivot soient nulles.

Les nouvelles lignes se calcule de la manière suivante :

$New\_E_{ij} = E_{ij} - E_{ic} * (New\_E_{lj})_j$

New_ligne 1 =(3 2 1 0 0 1800)- **2*** (0 1 0 0 1 600)
=(3 0 1 0 -2 600)

New_ligne 2 = (1 0 0 1 0 400) - **0*** (0 1 0 0 1 600)
= (1 0 0 1 0 400)

New_ligne 4 = (-30 -50 0 0 0 0) -(- **50)*** (0 1 0 0 1 600)
= (-30 0 0 0 50 30000)

Ancien tableau

	x_1	x_2	x_3	x_4	x_5	b
x_3	3	**2**	1	0	0	1800
x_4	1	**0**	0	1	0	400
x_2	**0**	**1**	**0**	**0**	**1**	**600**
-Z	-30	-50	0	0	0	0

Programmation Linéaire

- **Méthode du Simplexe**
  - **Le Tableau du Simplexe.**

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 1800 \\ x_1 + x_4 = 400 \\ x_2 + x_5 = 600 \\ z - 30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Solution admissible (0,0)

Etape 7: test d'optimalité

La solution de base actuelle est x1=0 (car n'apparait pas dans les variables de base), x2=600, x3=600,x4=400, x5=0, avec Z=50*600=30000.

∃ Valeur négative, on continue

La nouvelle Table

	x_1	x_2	x_3	x_4	x_5	b
x_3	3	0	1	0	-2	600
x_4	1	0	0	1	0	400
x_2	0	1	0	0	1	600
Z	-30	0	0	0	50	30000

Programmation Linéaire

- **Méthode du Simplexe**
  - **Le Tableau du Simplexe.**

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 1800 \\ x_1 + x_4 = 400 \\ x_2 + x_5 = 600 \\ z - 30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Solution admissible (0,0)

Itération 2

Etape 4: Choix de la Variable Entrante

x_1 est la variable entrante

	x_1	x_2	x_3	x_4	x_5	b
x_3	3	0	1	0	-2	600
x_4	1	0	0	1	0	400
x_2	0	1	0	0	1	600
Z	-30	0	0	0	50	30000

Programmation Linéaire

- **Méthode du Simplexe**
  - **Le Tableau du Simplexe.**

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 1800 \\ x_1 + x_4 = 400 \\ x_2 + x_5 = 600 \\ z - 30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Solution admissible (0,0)

Itération 2

Étape 5: Choix de la Variable sortante

	x_1	x_2	x_3	x_4	x_5	b	bi/Eic
x_3	3	0	1	0	-2	600	200
x_4	1	0	0	1	0	400	400
x_2	0	1	0	0	1	600	+oo
Z	-30	0	0	0	50	30000	

La variable sortante x_3 qui sera remplace par x_1 .

Programmation Linéaire

- **Méthode du Simplexe**
  - **Le Tableau du Simplexe.**

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 1800 \\ x_1 + x_4 = 400 \\ x_2 + x_5 = 600 \\ z - 30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Itération 2

Étape 6: Pivotage, normalisation

	x_1	x_2	x_3	x_4	x_5	b
x_1	**3**	**0**	**1**	**0**	**-2**	600
x_4	**1**	0	0	1	0	400
x_2	0	1	0	0	1	600
Z	**-30**	0	0	0	50	30000

$$\begin{cases} E_{l1} = \frac{3}{3} = 1 \\ E_{l2} = \frac{0}{3} = 0 \\ E_{l3} = \frac{1}{3} \\ E_{l4} = \frac{0}{3} = 0 \\ E_{l5} = \frac{-2}{3} \end{cases}$$

Elc=3

Nouvelle ligne pivot

	x_1	x_2	x_3	x_4	x_5	b
x_1	**1**	0	1/3	0	-2/3	200
x_4	**1**	0	0	1	0	400
x_2	**0**	1	0	0	1	600
Z	**-30**	0	0	0	50	30000

Programmation Linéaire

- **Méthode du Simplexe**
  - **Le Tableau du Simplexe.**

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 1800 \\ x_1 + x_4 = 400 \\ x_2 + x_5 = 600 \\ z - 30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Itération 2

Étape 6: Pivotage, normalisation

	x_1	x_2	x_3	x_4	x_5	b
x_1	**1**	**0**	**1/3**	**0**	**-2/3**	**200**
x_4	1	0	0	1	0	400
x_2	0	1	0	0	1	600
Z	**-30**	0	0	0	50	30000

Les nouvelles lignes se calcule de la manière suivante :

$New\_E_{ij} = E_{ij} - E_{ic} * (New\_E_{lj})_j$

New_ligne 2 = (1 0 0 1 0 400)- **1*** (1 0 1/3 0 -2/3 200) =(0 0 -1/3 1 2/3 600)

New_ligne 3 = (0 1 0 0 1 400) - **0*** (1 0 1/3 0 -2/3 200) = (0 1 0 0 1 600)

New_ligne 4 = (-30 0 0 0 50 30000) + **30*** (1 0 1/3 0 -2/3 200) = (0 0 10 0 30 36000)

Programmation Linéaire

▪ Méthode du Simplexe

- **Le Tableau du Simplexe.**

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 1800 \\ x_1 + x_4 = 400 \\ x_2 + x_5 = 600 \\ z - 30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Itération 2

Étape 7: Test

	x_1	x_2	x_3	x_4	x_5	b
x_1	**1**	0	**1/3**	**0**	**-2/3**	**200**
x_4	0	0	-1/3	1	2/3	200
x_2	0	1	0	0	1	600
Z	0	0	10	0	30	36000

- La solution de base actuelle est : x_1 =200, x_2 =600, x_3 =0, x_4=200, x_5 =0, avec **Z=36000.**
- Tous les coefficients des variables hors base dans la ligne Z sont **non négatifs** (10 et 30). Cela signifie que la solution actuelle est optimale.
- Pour **maximiser** le profit Z, l'entreprise doit produire **200 unités** du produit x1 et **600** unités du produit x2, ce qui générera un profit maximal de **36000**.

Programmation Linéaire

▪ Méthode du Simplexe

- **Le Tableau du Simplexe.**

$$\begin{cases} 3x_1 + 2x_2 + {\color{red}x_3} = 1800 \\ x_1 + {\color{red}x_4} = 400 \\ x_2 + {\color{red}x_5} = 600 \\ z - 30x_1 - 50x_2 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Itération 2

Étape 7: Test

	x_1	x_2	x_3	x_4	x_5	b
x_1	1	0	1/3	0	-2/3	200
x_4	0	0	-1/3	1	2/3	200
x_2	0	1	0	0	1	600
Z	0	0	10	0	30	36000

- Les variables d'écart nous indiquent :
  - ✓ $\boldsymbol{x_3}$ =0: La première contrainte (3 $\boldsymbol{x_1}$ + $\boldsymbol{2x_2}$ ≤1800) **est saturée,** entièrement utilisée.
  - ✓ x_4 =200: La deuxième contrainte (x_1 ≤400) a une marge **de 200 unités non utilisées.**
  - ✓ x_5 =0: La troisième contrainte (x_2 ≤600) **est saturée** (entièrement utilisée).

Programmation Linéaire

▪ Méthode du Simplexe

Cas particuliers : Plusieurs cas peuvent se présenter lors d'une résolution d'un problème PL.

- **Solutions multiples** : Si, à l'optimalité, une variable hors base a un coefficient de zéro dans la ligne Z, cela indique qu'il existe d'autres solutions optimales avec la même valeur de Z.
- **Problème non borné** : Si, lors du choix de la variable sortante, tous les coefficients de la **variable entrante** dans les contraintes sont non positifs, cela signifie que la variable entrante peut augmenter indéfiniment sans violer les contraintes, et la fonction objective peut croître (ou décroître pour la minimisation) sans limite. Le problème est non borné.
  - ✓ Si $z=x1+x2$ avec $x1-x2\leq 1$ et $x1,x2\geq 0$ alors x2 peut augmenter infiniment

Programmation Linéaire

▪ Méthode du Simplexe

Cas particuliers : Plusieurs cas peuvent se présenter lors d'une résolution d'un problème PL.

- **Dégénérescence** : Si une variable de base a une valeur bi zéro, le test du ratio minimum peut conduire à un ratio de zéro, ce qui peut potentiellement **entraîner des cycles** (l'algorithme revient à un tableau déjà rencontré). Des techniques comme la **règle de Bland** peuvent être utilisées pour éviter le cyclage.
  - ✓ **Règle de Bland** :
    - Choisir toujours la variable entrante avec le plus petit indice.
    - En cas d'égalité dans le ratio, choisir la variable sortante avec le plus petit indice.
- **Absence de solution réalisable** : Les contraintes définissent une région admissible vide. Il n'existe aucun ensemble de valeurs pour les variables de décision qui satisfassent simultanément toutes les contraintes du problème.

Programmation Linéaire

- **Méthode du Simplexe**
  - L'algorithme du simplexe nécessite une **solution de base réalisable initiale** pour démarrer ses itérations.
  - Si le problème au format standard ne présente pas immédiatement une base **réalisable évidente**, des méthodes spéciales sont nécessaires pour en trouver une.
    - ✓ Les deux méthodes principales sont **la phase I** et **le grand M,** qui ne seront pas détaillées ici.
  - En plus de la méthode du simplexe classique, il existe d'autres versions basées sur le simplexe et d'autres algorithmes pour résoudre des problèmes de PL.

Programmation Linéaire

- **Méthode du Simplexe**
  - Méthode du Simplexe Dual
  - Méthode du Simplexe Borné (Bounded Variable Simplexe)
  - Méthode du Simplexe Révisé (Revised Simplexe Method)

Programmation Linéaire

- **Autres Algorithmes de Programmation Linéaire**
  - Algorithmes Spécialisés pour des problèmes particulières:
    - ✓ **Algorithme de Transport** : Pour les problèmes de distribution de biens d'un ensemble de sources (fournisseurs) à un ensemble de destinations (clients), en minimisant les coûts logistiques.
    - ✓ **Algorithme d'Affectation** (**Méthode Hongroise**) : Affectation optimale de ressources (employés, machines) à des tâches, avec des coûts ou temps variables.
    - ✓ **Algorithmes de Flot dans les Réseaux** (**Ford-Fulkerson, Edmonds-Karp**) : Optimisation de trafic routier, réseaux de télécommunication, gestion d'énergie.
    - ✓ **Décomposition de Dantzig-Wolfe** : Pour les problèmes de grande taille avec une structure bloc-diagonale: Planification multi-étapes, problèmes de découpe de matériaux.
  - Logiciels de Programmation Linéaire : solveurs commerciaux et open source (**CPLEX**, Gurobi, **SciPy.optimize** ou **cvxpy**).

Programmation Linéaire

- **Méthode du Simplexe**
  - **Avantages**
    - ✓ Très performante pour la plupart des problèmes réels de programmation linéaire (souvent converge en $O(n)$ à $O(n^3)$ itérations).
    - ✓ Optimise rapidement des problèmes avec des milliers de variables/contraintes.
    - ✓ Fournit une solution **optimale exacte** globale (pas une approximation) lorsque le problème est **borné** et réalisable.
    - ✓ Interprétation économique : Les variables duales (écart et excédent) donnent des insights utiles (valeur marginale des ressources, sensibilité aux contraintes).

Programmation Linéaire

- **Méthode du Simplexe**
  - **limites**
    - ✓ Dans le pire cas la complexité théorique exponentielle l'algorithme peut explorer tous les sommets du polyèdre $O(2^n)$ itérations.
    - ✓ Sensibilité aux dégénérescences : Risque de cyclage (boucles infinies) si les règles anti-cyclage comme la règle de Bland ne sont pas appliquées, ce qui peut ralentir la convergence.
    - ✓ Nécessite une reformulation préalable en ajoutant les variables d'écart/excédents.
    - ✓ Limite aux problèmes linéaires : Inadaptée aux problèmes non-linéaires ou discrets.