

# Programmation Linéaire

## Introduction

Fabian Bastin

`fabian.bastin@umontreal.ca`

Université de Montréal – CIRRELT – IVADO – Fin-ML



# Objectifs du cours

1. Bases de la programmation mathématique :
  - objectif, contraintes ;
  - solution(s) réalisable(s) ;
  - solution(s) optimale(s), valeur optimale.
2. Comprendre les principes de base de la programmation linéaire (PL), et de ses méthodes de résolution. La méthode du simplexe sera étudiée avec un niveau élevé de détails.
3. Comprendre la notion de dualité.
4. Principes de base des méthodes de points intérieurs.
5. Introduction à la programmation en nombres entiers.
6. Aspects numériques de la PL (utilisation de Julia).

## Ouvrages de références

- Luenberger et Ye, Linear and Nonlinear Programming, 5e édition, Springer, 2021, <https://link.springer.com/book/10.1007/978-3-030-85450-8>
- Chan, Batson, Dang, Applied Integer Programming : Modeling and Solution, Wiley, 2011, <https://onlinelibrary.wiley.com/doi/book/10.1002/9781118166000>

# Modalités

## Évaluations :

- 4 devoirs, 10% chaque ;
- Examen intra : 25%, une page de notes autorisée ;
- Examen final : 35%, deux pages de notes autorisées.

**Le règlement sur le plagiat sera de stricte application !**

Démonstrateur : Pierre Borie `jpierre.borie@umontreal.ca`

- Notebooks : <https://github.com/fbastin/optim>;
- Slides : [https://github.com/fbastin/LP\\_Introduction](https://github.com/fbastin/LP_Introduction);
- Canal discord : <https://discord.gg/KSjzR2ht4f>.

## PL : forme standard

$$\min_x z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

$$\text{sujet à } a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$\vdots \quad \quad \vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m$$

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$$

# Ecriture matricielle

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = \mathbf{c}^T \mathbf{x} \\ \text{sujet à} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

## Retour à la forme standard

### Variables d'écart

$$\begin{aligned} a_{j1}x_1 + a_{j2}x_2 + \dots + a_{jn}x_n &\leq b_j \\ \rightarrow a_{j1}x_1 + a_{j2}x_2 + \dots + a_{jn}x_n + s_j &= b_j \end{aligned}$$

### Variables de surplus

$$\begin{aligned} a_{j1}x_1 + a_{j2}x_2 + \dots + a_{jn}x_n &\geq b_j \\ \rightarrow a_{j1}x_1 + a_{j2}x_2 + \dots + a_{jn}x_n - u_j &= b_j \end{aligned}$$

## Retour à la forme standard (suite)

### Variables libres

$x_j$  non contraint :  ~~$x_j \geq 0$~~

### Première approche : décomposition

$$x_j = s_j - u_j, \quad s_j \geq 0, \quad u_j \geq 0.$$

### Deuxième approche : substitution

Supposons  $a_{kj} \neq 0$

$$x_j = \frac{b_k}{a_{kj}} - \frac{a_{k1}}{a_{kj}}x_1 - \frac{a_{k2}}{a_{kj}}x_2 - \dots - \frac{a_{k,j-1}}{a_{kj}}x_{j-1} - \frac{a_{k,j+1}}{a_{kj}}x_{j+1} - \dots - \frac{a_{kn}}{a_{kj}}x_n.$$

On remplace dans les autres contraintes.



# Maximisation vs minimisation

Le problème

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbf{c}^T \mathbf{x} \\ \text{sujet à} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

est équivalent à

$$\begin{aligned} - \min_{\mathbf{x}} \quad & -\mathbf{c}^T \mathbf{x} \\ \text{sujet à} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

## Réalisabilité – optimalité

$\mathbf{x}$  est une solution réalisable si  $\mathbf{Ax} = \mathbf{b}$  et  $\mathbf{x} \geq 0$ .

$\mathbf{x}$  est une solution optimale si  $\mathbf{x}$  est réalisable et il n'existe pas de  $\mathbf{x}'$  réalisable tel que  $\mathbf{c}^T \mathbf{x}' < \mathbf{c}^T \mathbf{x}$ .

Supposons  $m \leq n$  et  $\text{rang}(\mathbf{A}) = m$ . Sans perte de généralité, supposons les  $m$  premières colonnes de  $\mathbf{A}$  indépendantes, et formons

$$\mathbf{A} = (\mathbf{B} \quad \mathbf{D})$$

Solution de base :  $\mathbf{x} = (\mathbf{x}_b \ 0)$ , avec  $\mathbf{B}\mathbf{x}_b = \mathbf{b}$ .

## Exemple

*Une compagnie produit des téléphones intelligents et des tablettes électroniques. Elle dispose à cette fin de trois usines (usine 1, usine 2, usine 3), qui ont chacune une capacité de production limitée. Les coques et modules de communication des téléphones sont produits dans l'usine 1, les coques et composants électroniques des tablettes dans l'usine 2, et l'usine 3 produit les écrans et assemble les produits. La compagnie a décidé de mettre en place de ligne de production :*

- *produit 1 : un téléphone ;*
- *produit 2 : une tablette.*

*Un lot de 20 unités donne lieu à un profit de \$3000 et \$5000, respectivement pour le produit 1 et le produit 2.*

## Exemple

|                | <i>Produit 1</i><br><i>Tps de prod. (h)</i> | <i>Produit 2</i><br><i>Tps de prod. (h)</i> | <i>Capacité de</i><br><i>production (h)</i> |
|----------------|---|---|---|
| <i>Usine 1</i> | 1   | 0   | 4   |
| <i>Usine 2</i> | 0   | 2   | 12  |
| <i>Usine 3</i> | 3   | 2   | 18  |

*Chaque lot d'un produit est le résultat combiné de la production dans les trois usines. Nous souhaitons déterminer le taux de production pour chaque produit (nombre de lots par semaine) de façon à maximiser le profit total.*

*Les variables de décision sont*

- $x_1$ , le nombre de lots du produit 1 ;
- $x_2$ , le nombre de lots du produit 2.

*La fonction objectif est le profit total, qui vaut  $3x_1 + 5x_2$ , en l'exprimant en millier de dollars. Nous voulons maximiser ce profit.*

## Exemple

*En résumé, nous avons le problème d'optimisation suivant :*

$$\max_x z = 3x_1 + 5x_2$$

$$t.q. \quad x_1 \leq 4 \quad (\text{usine 1})$$

$$2x_2 \leq 12 \quad (\text{usine 2})$$

$$3x_1 + 2x_2 \leq 18 \quad (\text{usine 3})$$

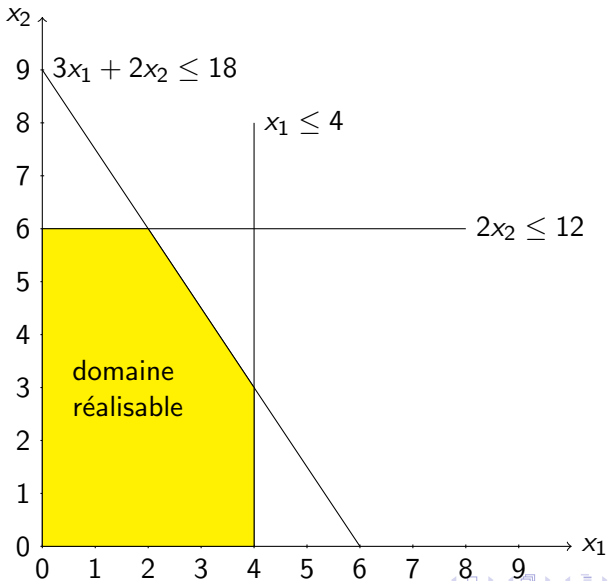
$$x_1 \geq 0, \quad x_2 \geq 0 \quad (\text{non-négativité})$$

*Dans l'exemple*

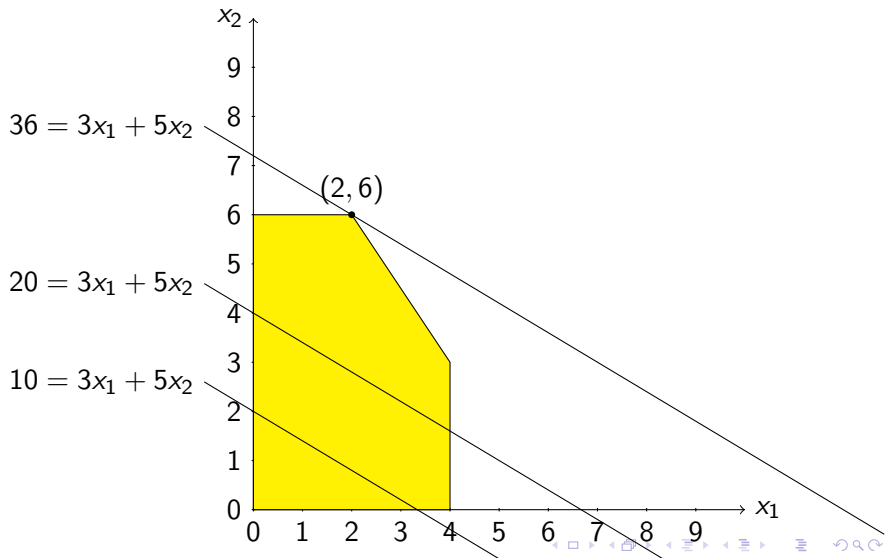
$$z = 3x_1 + 5x_2 \Leftrightarrow x_2 = -\frac{3}{5}x_1 + \frac{1}{5}z.$$

*L'ordonnée à l'origine, dépendant de la valeur de  $z$ , est  $\frac{1}{5}z$ , et la pente vaut  $-\frac{3}{5}$ . Maximiser revient à augmenter  $z$ .*

## Solution graphique



## Solution graphique



# JuMP

JuMP : Julia for Mathematical Optimization

```
using JuMP  
using HiGHS
```

```
m = Model(HiGHS.Optimizer)
```

```
@variable(m, 0 <= x <= 4)
```

```
@variable(m, 0 <= y <= 6)
```

```
@constraint(m, 3x + 2y <= 18.0)
```

```
@objective(m, Max, 3x+5y)
```

```
status = optimize!(m)
```