# HAI722I — DM

Ivan Lejeune

10 décembre 2025

## Table des matières

## Instructions

Ce devoir est à rendre avant le 12 décembre 2025 à 12h, soit par mail à l'adresse : rodolphe.giroudeau@lirmm.fr, soit en déposant votre devoir durant le cours

# 1 Partie théorique

**Exercice 1 Algorithmes pour la programmation linéaire.**
Considérons la formulation suivante :

$$P_\beta = \begin{cases} \max z = 5x_1 + 2x_2 \\ 6x_1 + x_2 \geq 6 \\ 4x_1 + 4x_2 \geq 12 \\ x_1 + 2x_2 \geq 4 \\ x_i \geq 0, \quad \forall i \in \{1,2\} \end{cases}$$

1. Résoudre le problème $P_\beta$ par la méthode du big $M$.
2. Résoudre le problème $P_\beta$ par la méthode à deux phases.
3. **Difficile :**
   (a) Résoudre le problème $P_\beta$ par la méthode dual-simplexe.
   (b) Soit le programme linéaire $P_\theta$

$$P_\theta = \begin{cases} \max z = x_1 + 3x_2 \\ x_1 + x_2 \geq 3 \\ x_1 - 2x_2 \geq 5 \\ -2x_1 + x_2 \leq 5 \\ x_i \geq 0, \quad \forall i \in \{1,2\} \end{cases}$$

Résoudre le problème $P_\theta$ par la méthode dual-simplexe.

**Solution.**

1. Commençons par poser le problème sous forme standard :

$$P_\beta = \begin{cases} \max z = 5x_1 + 2x_2 + 0x_3 + 0x_4 + 0x_5 - M \cdot (y_1 + y_2 + y_3) \\ 6x_1 + x_2 - x_3 + y_1 = 6 \\ 4x_1 + 4x_2 - x_4 + y_2 = 12 \\ x_1 + 2x_2 - x_5 + y_3 = 4 \\ x_i \geq 0, y_j \geq 0, \quad \forall i \in \{1,\dots,5\}, \forall j \in \{1,2,3\}. \end{cases}$$

Ensuite on construit notre tableau du simplexe :

| | | $c$ | 5 | 2 | 0 | 0 | 0 | $-M$ | $-M$ | $-M$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $c^J$ | variables de base | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $y_1$ | $y_2$ | $y_3$ |
| $-M$ | $x_1^1 = y_1$ | 6 | 6 | 1 | $-1$ | 0 | 0 | 1 | 0 | 0 |
| $-M$ | $x_2^1 = y_2$ | 12 | 4 | 4 | 0 | $-1$ | 0 | 0 | 1 | 0 |
| $-M$ | $x_3^1 = y_3$ | 4 | 1 | 2 | 0 | 0 | $-1$ | 0 | 0 | 1 |
| | $z(x)$ | $-22M$ | $-11M-5$ | $-7M-2$ | $M$ | $M$ | $M$ | 0 | 0 | 0 |

et on déroule l'algorithme :

- on rentre $x_1$,

- on sort $y_1$ car $1 < 3 < 4$,

$\downarrow$

| | | $c$ | 5 | 2 | 0 | 0 | 0 | $-M$ | $-M$ | $-M$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $c^J$ | variables de base | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $y_1$ | $y_2$ | $y_3$ |
| 5 | $x_1^2 = x_1$ | 1 | 1 | $\frac{1}{6}$ | $-\frac{1}{6}$ | 0 | 0 | $\frac{1}{6}$ | 0 | 0 |
| $-M$ | $x_2^2 = y_2$ | 8 | 0 | $\frac{10}{3}$ | $\frac{4}{6}$ | $-1$ | 0 | $-\frac{4}{6}$ | 1 | 0 |
| $-M$ | $x_3^2 = y_3$ | 3 | 0 | $\frac{11}{6}$ | $\frac{1}{6}$ | 0 | $-1$ | $-\frac{1}{6}$ | 0 | 1 |
| | $z(x)$ | $-11M+5$ | 0 | $-\frac{31}{6}M - \frac{7}{12}$ | $-\frac{5}{6}M - \frac{5}{6}$ | $M$ | $M$ | $\frac{11}{6}M + \frac{5}{6}$ | 0 | 0 |

- on rentre $x_2$,
- on sort $y_3$ car $\frac{11}{2} < 6 < \frac{80}{3}$,

$\downarrow$

| | | $c$ | 5 | 2 | 0 | 0 | 0 | $-M$ | $-M$ | $-M$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $c^J$ | variables de base | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $y_1$ | $y_2$ | $y_3$ |
| 5 | $x_1^3 = x_1$ | $\frac{8}{11}$ | 1 | 0 | $-\frac{2}{11}$ | 0 | $\frac{1}{11}$ | $\frac{2}{11}$ | 0 | $-\frac{1}{11}$ |
| $-M$ | $x_2^3 = y_2$ | $\frac{28}{11}$ | 0 | 0 | $\frac{4}{11}$ | $-1$ | $\frac{20}{11}$ | $-\frac{4}{11}$ | 1 | $-\frac{20}{11}$ |
| 2 | $x_3^3 = x_2$ | $\frac{18}{11}$ | 0 | 1 | $\frac{1}{11}$ | 0 | $-\frac{6}{11}$ | $-\frac{1}{11}$ | 0 | $\frac{6}{11}$ |
| | $z(x)$ | $-\frac{28}{11}M + \frac{76}{11}$ | 0 | 0 | $-\frac{4}{11}M - \frac{8}{11}$ | $-M$ | $-\frac{20}{11}M - \frac{7}{11}$ | $\frac{15}{11}M + \frac{8}{11}$ | 0 | $\frac{31}{11}M + \frac{7}{11}$ |

- on rentre $x_5$,
- on sort $y_2$ car $\frac{7}{5} < 8$ (l'autre rapport étant négatif on le compte pas),

$\downarrow$

| | | $c$ | 5 | 2 | 0 | 0 | 0 | $-M$ | $-M$ | $-M$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $c^J$ | variables de base | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $y_1$ | $y_2$ | $y_3$ |
| 5 | $x_1^4 = x_1$ | $\frac{3}{5}$ | 1 | 0 | $-\frac{1}{5}$ | $\frac{1}{20}$ | 0 | $\frac{1}{5}$ | $-\frac{1}{20}$ | 0 |
| 0 | $x_2^4 = x_5$ | $\frac{7}{5}$ | 0 | 0 | $\frac{1}{5}$ | $-\frac{11}{20}$ | 1 | $-\frac{1}{5}$ | $\frac{11}{20}$ | $-1$ |
| 2 | $x_3^4 = x_2$ | $\frac{12}{5}$ | 0 | 1 | $\frac{1}{5}$ | $-\frac{3}{10}$ | 0 | $-\frac{1}{5}$ | $\frac{3}{10}$ | 0 |
| | $z(x)$ | $\frac{39}{5}$ | 0 | 0 | $-\frac{3}{5}$ | $-\frac{7}{20}$ | 0 | $M + \frac{3}{5}$ | $M + \frac{7}{20}$ | $M$ |

A ce stade, il est clair que les variables artificielles ne pourront plus entrer en base. On peut donc les retirer et procéder sur le tableau réduit suivant :

| | | | $c$ | 5 | 2 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|
| $c^J$ | variables de base | | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
| 5 | $x_1^4 = x_1$ | $\frac{3}{5}$ | | 1 | 0 | $-\frac{1}{5}$ | $\frac{1}{20}$ | 0 |
| 0 | $x_2^4 = x_5$ | $\frac{7}{5}$ | | 0 | 0 | $\frac{1}{5}$ | $-\frac{11}{20}$ | 1 |
| 2 | $x_3^4 = x_2$ | $\frac{12}{5}$ | | 0 | 1 | $\frac{1}{5}$ | $-\frac{3}{10}$ | 0 |
| | $z(x)$ | $\frac{39}{5}$ | | 0 | 0 | $-\frac{3}{5}$ | $-\frac{7}{20}$ | 0 |

- on rentre $x_3$,
- on sort $x_5$,

$\downarrow$

| | | | $c$ | 5 | 2 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|
| $c^J$ | variables de base | | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
| 5 | $x_1^5 = x_1$ | 2 | | 1 | 0 | 0 | $-\frac{1}{2}$ | 1 |
| 0 | $x_2^5 = x_3$ | 7 | | 0 | 0 | 1 | $-\frac{11}{4}$ | 5 |
| 2 | $x_3^5 = x_2$ | 1 | | 0 | 1 | 0 | $\frac{1}{4}$ | $-1$ |
| | $z(x)$ | 12 | | 0 | 0 | 0 | $-2$ | 3 |

Il ne reste probablement qu'une étape, faisons-la :

- on rentre $x_4$,
- on sort $x_2$, c'est la seule valeur positive,

$\downarrow$

| | | | $c$ | 5 | 2 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|
| $c^J$ | variables de base | | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
| 5 | $x_1^6 = x_1$ | 4 | | 1 | 2 | 0 | 0 | $-1$ |
| 0 | $x_2^6 = x_3$ | 18 | | 0 | 11 | 1 | 0 | $-6$ |
| 0 | $x_3^6 = x_4$ | 4 | | 0 | 4 | 0 | 1 | $-4$ |
| | $z(x)$ | 20 | | 0 | 8 | 0 | 0 | $-5$ |

Dommage, on a encore un point pivot à faire rentrer ($x_5$ a valeur négative) donc on aimerait continuer de dérouler l'algorithme mais on a pas de critère pour trouver quelle variable sortir de base (en effet, toutes les valeurs dans la colonne du pivot sont négatives).

On conclut donc que l'espace des solutions est non borné. C'est-à-dire qu'il n'existe pas « un couple maximum » car on peut toujours en trouver un plus grand. Donc il n'y a pas un unique couple $(x_1, x_2)$ qui maximise $z$.

2. Résolvons maintenant le problème $P_\theta$ par la méthode à deux phases.

**Exercice 2 Dualité.**

Considérez le programme linéaire le plus général envisageable donné ci-dessous :

$$\begin{cases} \min z = c_1 x_1 + c_2 x_2 \\ A_{11} x_1 + A_{12} x_2 \leq b_1 \\ A_{21} x_1 + A_{22} x_2 = b_2 \\ x_i \geq 0, \quad \forall i \in \{1, 2\} \end{cases}$$

où $A$ est une matrice $(m_1 + m_2) \times (n_1 + n_2)$ et $c, x \in \mathbb{R}^{n_1+n_2}$ et $b \in \mathbb{R}^{m_1+m_2}$.
Caractériser le dual.

**Solution.**

**Exercice 3 Ensemble convexe.**
Soit $C_1$ et $C_2$ deux convexes de $\mathbb{R}^{m+n}$. Montrer que l'ensemble

$$C = \{(x, y_1 + y_2) \mid x \in \mathbb{R}^m, y_1 \in \mathbb{R}^n, y_2 \in \mathbb{R}^n, (x, y_1) \in C_1, (x, y_2) \in C_2\}$$

est également convexe.

**Solution.** Soient $(x^1, y^1)$ et $(x^2, y^2)$ deux points de $C$. Par définition de $C$, il existe $y_1^1, y_2^1, y_1^2, y_2^2$ tels que

$$(x^1, y_1^1) \in C_1, \quad (x^1, y_2^1) \in C_2, \quad (x^2, y_1^2) \in C_1, \quad (x^2, y_2^2) \in C_2,$$

et

$$y^1 = y_1^1 + y_2^1, \quad y^2 = y_1^2 + y_2^2.$$

Soit $\lambda \in [0, 1]$. Considérons le point

$$(x^\lambda, y^\lambda) = \lambda(x^1, y^1) + (1 - \lambda)(x^2, y^2).$$

On a

$$\begin{aligned} x^\lambda &= \lambda x^1 + (1 - \lambda) x^2, \\ y^\lambda &= \lambda y^1 + (1 - \lambda) y^2 \\ &= \lambda(y_1^1 + y_2^1) + (1 - \lambda)(y_1^2 + y_2^2) \\ &= (\lambda y_1^1 + (1 - \lambda) y_1^2) + (\lambda y_2^1 + (1 - \lambda) y_2^2). \end{aligned}$$

Notons $y_1^\lambda = \lambda y_1^1 + (1 - \lambda) y_1^2$ et $y_2^\lambda = \lambda y_2^1 + (1 - \lambda) y_2^2$. Par convexité de $C_1$ et $C_2$, on a

$$(x^\lambda, y_1^\lambda) \in C_1, \quad (x^\lambda, y_2^\lambda) \in C_2.$$

Ainsi, par définition de $C$, on a $(x^\lambda, y^\lambda) \in C$. Donc $C$ est convexe.

**Exercice 4 Modélisation et dualité.**
Considérons un problème d'affectation avec $m$ jobs et $n$ travailleurs ($n \geq m$). Chaque job doit être affecté à exactement un travailleur. Soit $p_{ij}$ le rendement obtenu si on affecte le job $i$ au travailleur $j$, où $i \in \{1, \ldots, m\}$ et $j \in \{1, \ldots, n\}$. On cherche une affectation qui maximise le rendement total.

1. Donner le programme linéaire.
2. Donner la formulation du dual de ce problème.

**Solution.**

1. On introduit les variables $x_{ij}$ qui valent 1 si le job $i$ est affecté au travailleur $j$, et 0 sinon. Le programme linéaire s'écrit alors :

$$P_0 = \begin{cases} \max z = \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} & (1) \\ \sum_{j=1}^{n} x_{ij} = 1, \quad \forall i \in \{1,\dots,m\} & (2) \\ \sum_{i=1}^{m} x_{ij} \leq 1, \quad \forall j \in \{1,\dots,n\} & (3) \\ x_{ij} \in \{0,1\}, \quad \forall i,j & (4) \end{cases}$$

où les contraintes correspondent à :

(1) on maximise le rendement total
(2) chaque job est affecté à un travailleur
(3) chaque travailleur a au plus un job
(4) les variables sont 0 ou 1

2. Pour obtenir la formulation du dual, on introduit les variables duales $u_i$ (une par job) et $v_j$ (une par travailleur). Le dual s'écrit alors :

$$D_0 = \begin{cases} \min w = \sum_{i=1}^{m} u_i + \sum_{j=1}^{n} v_j & (5) \\ u_i + v_j \geq p_{ij}, \quad \forall i,j & (6) \\ v_j \geq 0, \quad \forall j & (7) \end{cases}$$

où les contraintes correspondent à :

(5) on minimise la somme des coûts
(6) les coûts doivent couvrir les rendements
(7) les variables sont positives

**Exercice 5 Programmation linéaire : Farkas.**
Considérons le programme linéaire suivant, qui dépend de $\varepsilon \in \mathbb{R}$ :

$$\begin{cases} \min z = 4x_1 - 2x_2 \\ x_2 \leq 3 \\ \varepsilon x_1 + (2-\varepsilon)x_2 \leq 4 \\ x_i \geq 0, \quad \forall i \in \{1,2\} \end{cases}$$

1. Montrer que le problème est réalisable $\forall \varepsilon \in \mathbb{R}$.
2. Pour quelles valeurs de $\varepsilon$ la valeur optimale est-elle non bornée ?

**Solution.**

**Exercice 6 Résolution numérique.**
Résoudre le programme linéaire suivant par la méthode Primal-Dual :

$$\text{Primal} = \begin{cases} \min z(x_1,x_2,x_3) = 2x_1 + x_2 + 2x_3 + 8x_4 \\ 2x_1 - x_2 + 3x_3 - 2x_4 = 3 \\ -x_1 + 3x_2 - 4x_3 = 1 \\ x_i \geq 0, \quad \forall i \in \{1,2,3,4\} \end{cases}$$

**Solution.**

## 2 Partie pratique

**Exercice 7 Résolution d'un programme linéaire avec Julia.**

1. Dans un premier temps nous allons regarder quelques commandes de base en Julia.
   (a) Pour lancer Julia :

```
julia
```

   (b) Introduction à la syntaxe. La syntaxe est assez simple :

```
a = 5
b = 67.67e42
c = a + 3 * b ^ 4
m = "Bonjour"
d = sin(atan(3))
toi = "Rosa"
println("Bonjour, $toi !")
e1 = [12; 34; 45]
e2 = [12 34 45]
d = true
f = [12.0; 34; "Youhou !"]
B = [1 2; 1 3; 4 5]
```

   (c) les fonctions sont bien définies :

```
function add(a, b)
    return a + b
end

f(x) = 42 * x^3

add(4, 2)
f(a)
```

   Renseignements obligatoires sur la syntaxe de Julia peuvent être trouvés sur les pages suivantes :

   - `https://julialang.org/`
   - `https://zestedesavoir.com/articles/78/a-la-decouverte-de-julia/`

   (d) Le package `JuMP` et le solveur `Cbe JuMP` sont des langages de modélisation pour l'optimisation mathématique intégré dans Julia. Pour l'utiliser il faut d'abord ajouter le package associé :

```
using Pkg
Pkg.add("JuMP")
Pkg.add("Cbc")
```

   Julia est un langage de modélisation et ne possède pas un solveur de programmation mathématique intégré. `CPLEX` est un solveur de programmation mathématique de haute performance, puissant pour la programmation linéaire, la programmation en nombres entiers et la programmation quadratique. Mais le solveur `CPLEX` n'est pas gratuit et requiert une license (facile à obtenir pour les étudiants mais le système est bridé). Nou sutiliserons donc plutôt `Cbc` qui est un solveur de programmation mathématique open-source moins puissant intégré dans Julia. Dans la suite, nous

présentons les commandes pour utiliser `Cbc`. Pour utiliser un solveur à partir de Julia il faut ajouter le package associé.

(e) Création du modèle :

Utiliser les commandes présentées dans cette section pour définir dans Julia un modèle associé au programme linéaire $P_1$. Les variables $x_1$ et $x_2$ représentent respectivement la quantité de deux types différents de yaourt produits. La fonction objectif maximise le profit de la production :

```
Z = max 4x + 5y
2x+y <= 800
x+2y <= 700
y <= 300
x,y >= 0
```

(f) Packages :

```
using JuMP
using Cbc
```

(g) Définition du modèle :

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

(h) Définition des variables, exemples de définition de variables :

```
@variable(m, x) # No bounds
@variable(m, x >= lb) # Lower bound only (note: 'lb <= x' is not valid)
@variable(m, lb <= x <= ub) # Both lower and upper bounds
@variable(m, x == fixedval) # Fixed variable
@variable(m, x[1:M, 1:N]) # Un vecteur de variables
@variable(m, x[1:5], Bin) # Un vecteur de variabes binaires
```

Ajouter à votre modèle les contraintes de $P_1$.

(i) Pour résoudre le modèle, il faut exécuter :

```
optimize!(m)
```

La valeur de la solution optimale du modèle, le temps d'exécution et la solutoin optimale, peuvent etre récupérés en utilisant les commandes suivantes :

```
Objective value(m)
    solve time(m)
    value.(x)
    value(x[i])
```

Voici un exemple, en supposant que $x$ est le vecteur de variables, profit et weight sont des vecteurs de coefficients :

```
println("Objective is: ", objective_value(m))
println("Solve time = ", solve_time(m))
println("Solution is:")
for i = 1:5
    print("x[$i] = ", value(x[i]))
    println(", p[$i]/w[$i] = ", profit[i]/weight[i])
end
```

**Solution.**

**Exercice 8.**

**Solution.**

**Exercice 9.**

**Solution.**