SIO - Simulation et optimisation

Programmation Linéaire

17 janvier 2026

Table des matières

1 Introduction à la Programmation Linéaire

La programmation linéaire est une méthode mathématique utilisée pour optimiser une fonction linéaire, appelée fonction objectif, sous certaines contraintes également linéaires. Elle est largement utilisée dans divers domaines tels que la gestion des ressources, la logistique, la production, et bien d'autres.

1.1 Structure d'un problème

Un problème de programmation linéaire se compose généralement de trois éléments principaux :

- **Fonction objectif** : C'est la fonction que l'on cherche à maximiser ou minimiser. Elle est exprimée sous forme linéaire.
- **Variables de décision** : Ce sont les variables que l'on peut ajuster pour optimiser la fonction objectif.
- **Contraintes** : Ce sont les restrictions ou les limites imposées aux variables de décision, également exprimées sous forme linéaire.

1.2 Résolution graphique

Une des méthodes les plus simples pour résoudre un problème de programmation linéaire à deux variables est la méthode graphique. Cette méthode consiste à tracer les contraintes sur un graphique, puis à identifier la région faisable et à déterminer le point optimal. Pour déterminer le point optimal, nous devons faire glisser la fonction objectif parallèlement jusqu'à ce qu'elle atteigne le dernier point de la région faisable.

1.3 Résulats possibles

Lors de la résolution d'un problème de programmation linéaire, plusieurs résultats sont possibles :

- **Vide**: dans un tel cas, le problème est sans solution admissible et ne possède donc pas de solution optimale.
- **Borné** (et non vide): dans ce cas, le problème possède une solution optimale unique ou multiple.
- **Non borné**: dans ce cas, selon la fonction objectif choisie,
  - le problème peut posséder des solutions optimales;
  - il peut exister des solutions admissibles de valeur arbitrairement grande/petite (en cas de maximisation/minimisation). Dans une telle situation, on dit que le problème est non borné et que le PL n'admet pas de solution optimale (finie).

2 Forme canonique

Un problème de programmation linéaire est dit être en forme canonique s'il est exprimé de la manière suivante :

$$\begin{aligned} \max z = & \sum_{j=1}^{n} c_j x_j \\ \text{s.c.} \quad & \sum_{j=1}^{n} a_{\mathrm{ij}} x_j \le b_i \quad i = 1, ..., m \\ & x_j \ge 0 \qquad\qquad j = 1, ..., n \end{aligned}$$

Nous pouvons définir la forme canonique par les critères suivants :

- Problème de **maximisation**
- Toutes les variables sont **positives**
- Toutes les contraintes (sauf celle de s'assurer que les variables sont positives) sont du type $\le$

2.1 Expression matricielle

Un problème de programmation linéaire en forme canonique peut être exprimé de manière matricielle comme suit :

$$\begin{aligned} \text{Max } z = & \, 3x_1 + 2x_2 \\ \text{s.c.} \quad & 2x_1 + x_2 \le 100 \\ & x_1 + x_2 \le 80 \\ & x_1 \le 40 \\ & x_1, x_2 \ge 0 \end{aligned}$$

Nous pouvons le réecrire en utilisant les matrices suivantes:

- $\vec{c}$: vecteur des coefficients de la fonction objectif
- $\vec{x}$: vecteur des variables de décision
- $\vec{A}$: matrice des coefficients des contraintes
- $\vec{b}$: vecteur des termes constants des contraintes

$$\vec{c} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}, \vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \vec{A} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}, \vec{b} = \begin{bmatrix} 100 \\ 80 \\ 40 \end{bmatrix}$$

2.2 Règles de transormmation

Pour transformer un problème de programmation linéaire en forme canonique, nous pouvons appliquer les règles suivantes :

- Pour la fonction objectif, si nous cherchons à minimiser, nous pouvons appliquer la conversion suivante

$$\min f(x) = -\max(-f(x))$$

- Pour passer d'une contrainte de type $\ge$ à une contrainte de type $\le$, nous multiplions les deux côtés de l'inégalité par -1 et cela nous permet d'inverser le sens de l'inégalité.

$$a_1x_1 + ... + a_nx_n \ge b \iff -a_1x_1 - ... - a_nx_n \le -b$$

- Pour transformer une contrainte d'égalité en deux contraintes d'inégalités, nous pouvons utiliser la conversion suivante :

$$a_1x_1 + ... + a_nx_n = b \iff \begin{cases} a_1x_1 + ... + a_nx_n \le b \\ a_1x_1 + ... + a_nx_n \ge b \end{cases}$$

- Pour passer d'une inéquation à une équation, nous devons ajouter une variable décart non négative :

$$a_1x_1 + ... + a_nx_n \leq b \Longleftrightarrow \begin{cases} a_1x_1 + ... + a_nx_n + s = b \\ s \geq 0 \text{ avec } s \geq 0 \end{cases}$$

- Tout nombre réel peut être écrit comme la différence de deux nombres non négatifs :

$$x \in R \Longleftrightarrow \begin{cases} x = x^+ - x^- \\ x^+, x^- \geq 0 \end{cases}$$

Dans ce cas, nous devons remplacer chaque occurrence de x par $x^+ - x^-$ dans la fonction objectif et les contraintes.

- Si une variable est bornée inférieurement :

$$x \geq b \Longleftrightarrow \begin{cases} x = x' + b \\ x' \geq 0 \end{cases}$$

3 Technique de linéarisation de base

Certaines contraintes ou fonctions objectives ne sont pas linéaires. Pour les résoudre à l'aide de la programmation linéaire, nous devons les linéariser. Nous allobns voir comment le faire pour les fonctions minimum, maximum et valeur absolue.

3.1 Contrainte min/max

Pour linéariser une contraite de **minimum**, il faut qu'elle soit sous la forme suivante :

$$\min(\text{expr}_1, ..., \text{expr}_n) \geq b$$

Pour linéairser une contraite de **maximum**, il faut qu'elle soit sous la forme suivante :

$$\max(\text{expr}_1, ..., \text{expr}_n) \leq b$$

Ces deux opérations peuvent être vues comme un opérateur **AND** logique. Pour linéariser ces contraintes, nous devons les transformer en plusieurs contraintes linéaires comme suit :

- Pour la contrainte de minimum :

$$\min(\text{expr}_1, ..., \text{expr}_n) \geq b \iff \begin{cases} \text{expr}_1 \geq b \\ ... \\ \text{expr}_n \geq b \end{cases}$$

Donc toutes les expressions doivent être supérieures ou égales à b.

- Pour la contrainte de maximum :

$$\max(\text{expr}_1, ..., \text{expr}_n) \leq b \iff \begin{cases} \text{expr}_1 \leq b \\ ... \\ \text{expr}_n \leq b \end{cases}$$

Donc toutes les expressions doivent être inférieures ou égales à b.

3.2 Objectif min/max

Pour linéariser une fonction objective de **minimum**, il faut qu'elle soit sous la forme suivante :

$$\text{Min } z = \max(\text{expr}_1, ..., \text{expr}_n)$$

Pour linéairser une fonction objective de **maximum**, il faut qu'elle soit sous la forme suivante :

$$\text{Max } z = \min(\text{expr}_1, ..., \text{expr}_n)$$

Pour linéariser ces fonctions objectives, nous devons appliquer le processus suivant:

- Introduire une nouvelle variable t qui représentera la valeur de la fonction objective.

$$\begin{aligned} &\text{Min } z = t \\ &\text{s.c} \qquad t = \max\{\text{expr}_1, ..., \text{expr}_n\} \end{aligned}$$

- Grâce à cette nouvelle variable t, nous pouvons trasnformer l'égalité en une opération plus grand ou égal.

$$\begin{aligned} &\text{Min } z = t \\ &\text{s.c} \qquad t \geq \max\{\text{expr}_1, ..., \text{expr}_n\} \end{aligned}$$

Ensuite, nous pouvons linéariser ces objectifs en utilisant la même technique que pour les contraintes min/max.

3.2.1 Exemple

Considérons le problème d'optimisation suivant :

$$\begin{aligned} \text{Min } z &= \max(3x_1 - 2x_2, -x_1 + 4x_2) \\ \text{s.c.} \qquad & 5x_1 - 3x_2 \leq 6 \\ & x_1, x_2 \geq 0 \end{aligned}$$

En appliquant la technique de linéarisation, nous introduisons une nouvelle variable t et reformulons le problème comme suit :

$$\begin{aligned} \text{Min } z &= t \\ \text{s.c.} \qquad 5x_1 - 3x_2 &\leq 6 \\ t &\geq 3x_1 - 2x_2 \\ t &\geq -x_1 + 4x_2 \\ x_1, x_2 &\geq 0 \\ t &\in R \end{aligned}$$

Voici un problème de programmation linéaire en forme canonique.

3.3 Linéarisation d'une valeur absolue

3.3.1 Contrainte

La contrainte non linéaire

$$|\text{expr}_1| \leq \text{expr}_2 \quad \text{ou} \quad \text{expr}_1 \geq |\text{expr}_2|$$

est équivalente aux deux inéquations linéaires

$$-\text{expr}_2 \leq \text{expr}_1 \leq \text{expr}_2 \iff \begin{cases} \text{expr}_1 \leq \text{expr}_2 \\ \text{expr}_1 \geq -\text{expr}_2 \end{cases}$$

Les contraintes non linéaires

$$|\text{expr}_1| \geq \text{expr}_2 \quad \text{et} \quad |\text{expr}_1| = \text{expr}_2$$

ne sont pas linéarisables en général sans l'introduction de variables binaires supplémentaires.

3.3.2 Objectif

La fonction objective non linéaire

$$\text{Min } z = |\text{expr}_1|$$

peut-être récrit sous la forme équivalente:

$$\begin{aligned} \text{Min } z &= t \\ \text{s.c.} \qquad & t = |\text{expr}_1| \\ & t \geq 0 \end{aligned}$$

qui possède le même optimum que:

$$\begin{aligned} \text{Min } z &= t \\ \text{s.c.} \qquad & t \geq |\text{expr}_1| \\ & t \geq 0 \end{aligned}$$

et donc que sa linéarisation:

$$\begin{aligned} \text{Min } z &= t \\ \text{s.c.} \qquad & -t \leq \text{expr}_1 \leq t \\ & t \geq 0 \end{aligned}$$

4 Programme linéaires en nombres entiers

Un programme linéaire en nombres entiers (PLNE) est un programme linéaire où certaines variables ne peuvent prendre que des valeurs entières. On pourra distinguer:

- Les PLNE purs: **toutes** les variables sont contraintes à être entières.
- Les PL binaires: **toutes** les variables sont contraintes à être binaires (0 ou 1).
- Les PL mixtes: **certaines** variables sont contraintes à être entières, les autres pouvant être continues.

4.1 Modélisation avec des variables entières

Les variables entières sont souvent utilisées pour modéliser des situations où les décisions sont discrètes, telles que le nombre d'unités produites, le nombre de véhicules utilisés, ou la sélection de projets.

4.1.1 Variables discrètes

Pour modéliser une variable discrète qui peut prendre des valeurs dans un ensemble fini, nous pouvons utiliser des variables auxiliaires binaires. Considérons une variable x qui ne peut prendre que deux valeurs :

$$x = 0 \quad \text{ou} \quad x = b$$

Pour modéliser x, on introduit une variable auxiliaire binaire y et on pose

$$x = by \quad \text{avec } y \in \{0, 1\}$$

De manière générale, une variable **bivalente** se modèlise de la manière suivante :

$$x \in \{a, b\} \quad \Longleftrightarrow \quad \begin{cases} x = a + (b-a)y \\ y \in \{0, 1\} \end{cases}$$

Dans le cas où une variable doit pouvoir prendre plusieurs valeurs discrètes, nous pouvons modéliser cela comme ça :

$$x \in \{2, 5, 10\} \quad \Longleftrightarrow \quad \begin{cases} x = 2y_1 + 5y_2 + 10y_3 \\ y_1 + y_2 + y_3 = 1 \\ y_k \in \{0, 1\}, k = 1, 2, 3 \end{cases}$$

4.1.2 Fonction objectif avec coûts fixes

Il se peut arriver que la fonction objectif comporte des coûts fixes associés à l'activation de certaines variables.

$$\begin{aligned} \text{Min } & z = Ky + cx \\ \text{s.c. } & \ldots \\ & x \leq My \qquad (1) \\ & 0 \leq x \leq M \\ & y \in \{0, 1\} \end{aligned}$$

La contrainte (1) garantit que lorsque:

- $y = 0$, alors $x = 0$ (la variable n'est pas activée, donc pas de coût variable).
- $y = 1$, alors x peut prendre n'importe quelle valeur entre 0 et M (la variable est activée, donc le coût variable s'applique).
- Si $x > 0$, alors y doit être égal à 1 (si la variable est activée, le coût fixe s'applique).
- Si $x = 0$, alors y peut être égal à 0 ou 1 (si la variable n'est pas activée, le coût fixe ne s'applique pas).

4.1.3 Variables semi-continues

Une variable semi-continue est une variable qui peut soit être égale à zéro, soit prendre une valeur dans un intervalle continu. Imaginons un intervalle $[a, b]$:

$$x = 0 \quad \text{ou} \quad a \leq x \leq b$$

Pour gérer ce cas nous devons introduire 2 variables auxiliaires : une variable binaire y et une variable continue t.

$$y = \begin{cases} 1 \text{ si } x \in [a, b] \\ 0 \text{ si } x = 0 \end{cases}$$

et

$$0 \leq t \leq 1 \quad \text{servant à paramétrer l'intervalle } [a, b].$$

Nous pouvons alors modéliser x de la manière suivante :

$$x \in \{0\} \cup [a, b] \quad \Longleftrightarrow \quad \begin{cases} x = ay + (b - a)t & (1) \\ t \leq y & (2) \\ 0 \leq t \leq 1 & (3) \\ y \in \{0, 1\} & (4) \end{cases}$$

Cette approche garantit que :

- Si $y = 1$, la contrainte (2) devient redondante avec les contraintes de borne (3) et (1) devient $x = a + (b - a)t$. La variable x varie alors entre a et b lorsque t varie entre 0 et 1.
- Si $y = 0$, les contraintes (2) et (3) forcent $t = 0$ et (1) devient $x = 0$.
- Si $t > 0$, la contrainte (2) force $y = 1$ et (1) devient $x = a + (b - a)t$ et x varie entre a et b.
- Si $t = 0$, on peut avoir $y = 0$ et $x = 0$ ou $y = 1$ et $x = a$.

4.1.4 Contraintes disjonctives (OU logique)

Les contraintes disjonctives modélisent des situations où **au moins une** contrainte parmi plusieurs doit être satisfaite.

Considérons deux contraintes dont au moins une doit être satisfaite :

$$2x_1 + 3x_2 \leq 12 \quad \text{OU} \quad 3x_1 + x_2 \leq 9$$

Pour modéliser cette disjonction, on introduit une variable binaire y et un nombre très grand M :

$$\begin{cases} 2x_1 + 3x_2 \leq 12 + M(1 - y) \\ 3x_1 + x_2 \leq 9 + My \\ y \in \{0, 1\} \end{cases}$$

Analyse :

- Si $y = 1$: la première contrainte est active, la seconde devient $3x_1 + x_2 \leq 9 + M$ (toujours satisfaite).
- Si $y = 0$: la seconde contrainte est active, la première devient $2x_1 + 3x_2 \leq 12 + M$ (toujours satisfaite).

Généralisation : Pour satisfaire **au moins K contraintes parmi L**, on introduit une variable binaire y_i par contrainte et on reformule :

$$\begin{cases} \sum_{j=1}^{n} a_{\text{ij}} x_j \leq b_i + M(1 - y_i) & i = 1, ..., L \\ \sum_{i=1}^{L} y_i \geq K & \\ y_i \in \{0, 1\} & i = 1, ..., L \end{cases}$$

4.1.5 Valeurs absolues

Nous pouvons facilement linéariser une contrainte avec valeur absolue de type $\leq$ en utilisant des inéquations linéaires.

$$|\sum_{j=1}^{n} a_j x_j| \leq b \Longleftrightarrow -b \leq \sum_{j=1}^{n} a_j x_j \leq b$$

Exemple

$$|2x_1 + 3_{x_2}| \leq 12 \quad \Longleftrightarrow \quad \begin{cases} 2x_1 + 3x_2 \leq & 12 \\ 2x_1 + 3x_2 \geq & -12 \end{cases}$$

Pour linéariser une contrainte avec valeur absolue de type $\geq$, on utilise des variables binaires car elle implique une disjonction (OU logique).

La contrainte non linéaire

$$|\sum_{j=1}^{n} a_j x_j| \geq b$$

est équivalente à la disjonction

$$\sum_{j=1}^{n} a_j x_j \geq b \quad \text{OU} \quad \sum_{j=1}^{n} a_j x_j \leq -b$$

Pour modéliser cette disjonction, on introduit une variable binaire y et un nombre très grand M :

$$\begin{cases} -\sum_{j=1}^{n} a_j x_j \leq -b + M(1-y) \\ \sum_{j=1}^{n} a_j x_j \leq -b + My \\ y \in \{0,1\} \end{cases}$$

4.1.5.1 Exemple

La contrainte $|x_1 - 2x_2| \geq 5$ devient :

$$\begin{cases} -x_1 + 2x_2 \leq -5 + M(1-y) \\ x_1 - 2x_2 \leq -5 + My \\ y \in \{0,1\} \end{cases}$$

5 Résolution des programmes linéaires en nombres entiers

5.1 Relaxation linéaire

La **relaxation linéaire** (ou relaxation continue) d'un PLNE consiste à remplacer les contraintes d'intégralité par des contraintes de non-négativité.

Exemple :

$$\text{PLNE: } x_1, x_2 \in \mathbb{N} \quad \Rightarrow \quad \text{Relaxation: } x_1, x_2 \geq 0$$

Cette transformation convertit le PLNE en un PL classique en variables continues, plus facile à résoudre.

5.2 Domaine admissible

- **PL** : Le domaine admissible est un polyèdre convexe (polytope si borné). La solution optimale se trouve à un sommet.
- **PLNE** : L'ensemble des solutions admissibles est un ensemble **discret et fini** de points à coordonnées entières.

Résoudre un PLNE est un **problème d'optimisation combinatoire** : trouver la meilleure solution parmi un ensemble fini (mais généralement gigantesque) de solutions.

5.3 Cas favorables

Dans certains cas, la solution optimale de la relaxation linéaire est déjà entière :

- Si **tous les sommets** du domaine admissible de la relaxation linéaire sont entiers, résoudre le PLNE n'est pas plus difficile que résoudre sa relaxation linéaire.
- Exemple important : les **problèmes de transbordement** (réseaux de transport avec offres, demandes et capacités entières) garantissent toujours une solution optimale entière.

5.4 Arrondir la solution : une fausse bonne idée

Lorsque la solution optimale de la relaxation linéaire n'est pas entière, peut-on simplement l'arrondir ?

Deux difficultés majeures :

1. **Admissibilité** : Comment arrondir (vers le haut, vers le bas, au plus proche ?) tout en garantissant que la solution reste admissible ? Il n'existe pas de méthode générale.
2. **Optimalité** : Même si la solution arrondie est admissible, rien ne garantit qu'elle soit optimale pour le PLNE.

Exemple (sac à dos) :

$$\begin{aligned} \text{Max } z &= 13x_1 + 16x_2 + 7x_3 + 4x_4 \\ \text{s.c.} \quad & 6x_1 + 8x_2 + 4x_3 + 3x_4 \leq 12 \\ & x_1, x_2, x_3, x_4 \in \{0, 1\} \end{aligned}$$

Solution de la relaxation : $(x_1, x_2, x_3, x_4) = (1, 0.75, 0, 0)$ avec $z = 25$.

Si on arrondit x_2 vers le bas : $(1, 0, 0, 0)$ donne $z = 13$ (admissible mais sous-optimal).

La vraie solution optimale est : $(0, 1, 1, 0)$ avec $z = 23$.