

SIO - Simulation et optimisation

Programmation Linéaire

17 janvier 2026

Table des matières

1 Introduction à la Programmation Linéaire	1
1.1 Structure d'un problème	2
1.2 Résolution graphique	2
1.3 Résultats possibles	2
2 Forme canonique	2
2.1 Expression matricielle	3
2.2 Règles de transormmation	3
3 Technique de linéarisation de base	4
3.1 Contrainte min/max	5
3.2 Objectif min/max	5
3.2.1 Exemple	6
3.3 Linéarisation d'une valeur absolue	6
3.3.1 Contrainte	6
3.3.2 Objectif	6
4 Analyse de sensibilité	6
4.1 Sensibilité de la fonction objectif	7
4.1.1 Modification d'un coefficient	7
4.1.2 Intervalle de stabilité	7
4.1.3 Rapport de sensibilité dans GLPK	7
4.2 Sensibilité du second membre	9
4.2.1 Effet géométrique	9
4.2.2 Analyse similaire	9
4.2.3 Rapport de sensibilité dans GLPK	9
4.3 Utilité de l'analyse de sensibilité	10
5 Programme linéaires en nombres entiers	10
5.1 Programme linéaire binaire	11
5.1.1 Exemple : Sélection de projets	11
5.2 Modélisation avec des variables entières	11
5.2.1 Variables discrètes	12
5.2.2 Fonction objectif avec coûts fixes	12
5.2.3 Variables semi-continues	13
5.2.4 Contraintes disjonctives (OU logique)	13
5.2.5 Valeurs absolues	14
5.2.5.1 Exemple	14
6 Résolution des programmes linéaires en nombres entiers	14
6.1 Relaxation linéaire	15
6.2 Domaine admissible	15
6.3 Cas favorables	15
6.4 Arrondir la solution : une fausse bonne idée	15

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 Résultats 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 :

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

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 \leq

2.1 Expression matricielle

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

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

Nous pouvons le réécrire 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 transformation

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

$$a_1 x_1 + \dots + a_n x_n \geq b \iff -a_1 x_1 - \dots - a_n x_n \leq -b$$

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

$$a_1 x_1 + \dots + a_n x_n = b \iff \begin{cases} a_1 x_1 + \dots + a_n x_n \leq b \\ a_1 x_1 + \dots + a_n x_n \geq b \end{cases}$$

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

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

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

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

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 :

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

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 allons voir comment le faire pour les fonctions minimum, maximum et valeur absolue.

3.1 Contrainte min/max

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

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

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

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

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 :

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

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

- Pour la contrainte de maximum :

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

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 :

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

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

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

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

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

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

- Grâce à cette nouvelle variable t , nous pouvons transformer l'égalité en une opération plus grande ou égale.

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

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 :

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

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

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

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

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

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:

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

qui possède le même optimum que:

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

et donc que sa linéarisation:

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

4 Analyse de sensibilité

L'analyse de sensibilité étudie le comportement de la solution optimale d'un programme linéaire et de sa valeur lorsque l'on modifie légèrement les coefficients du problème (qui ne sont pas toujours connus avec une précision parfaite).

Deux questions principales sont abordées :

- **Analyse de la fonction objectif** : Comment se comporte la solution optimale et sa valeur lorsqu'on modifie un coefficient de la fonction objectif ?
- **Analyse du second membre** : Comment se comporte la solution optimale et sa valeur lorsqu'on modifie le second membre (terme constant) d'une contrainte ?

4.1 Sensibilité de la fonction objectif

Considérons le problème de Geppetto SA :

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

La solution optimale est : $x_1 = 20$ soldats, $x_2 = 60$ trains, avec $z = 180$ francs.

4.1.1 Modification d'un coefficient

Que se passe-t-il si on modifie le coefficient c_1 (actuellement 3) dans la fonction objectif ?

Géométriquement, cela modifie la pente des lignes de niveau de la fonction objectif :

- Si $c_1 = 2$: Les lignes de niveau deviennent $2x_1 + 2x_2 = \text{constante}$. On obtient deux sommets optimaux : $(20, 60)$ et $(0, 80)$, avec une infinité de solutions optimales entre ces points (valeur = 160).
- Si $c_1 = 3$: Solution unique $(20, 60)$ avec $z = 180$ (situation actuelle).
- Si $c_1 = 4$: Les lignes de niveau deviennent $4x_1 + 2x_2 = \text{constante}$. On obtient deux sommets optimaux : $(20, 60)$ et $(40, 20)$, avec une infinité de solutions optimales (valeur = 200).

4.1.2 Intervalle de stabilité

Principe général : Quand on modifie un coefficient de la fonction objectif, le sommet optimal actuel et la base associée restent optimaux tant que ce coefficient varie dans un intervalle $[a, b]$.

- Dans cet intervalle, la **solution optimale** (valeurs des variables) ne change pas.
- Seule la **valeur de la fonction objectif** est modifiée.
- Aux bornes de l'intervalle, on obtient généralement plusieurs solutions optimales.
- En dehors de l'intervalle, le sommet optimal change et on obtient un nouvel intervalle de stabilité.

Pour l'exemple : tant que $c_1 \in]2; 4[$, la solution optimale reste $(20, 60)$ et seule la valeur z varie entre 160 et 200.

4.1.3 Rapport de sensibilité dans GLPK

Dans GUSEK, ces informations sont obtenues via **Tools → Generate LP Sensitivity Analysis (CTRL+9)**.

La **deuxième page** du rapport contient l'analyse pour la fonction objectif :

GLPK 4.65 - SENSITIVITY ANALYSIS REPORT								Page	2	
Problem:		geppetto								
Objective:		z = 180 (MAXimum)								
No.	Column name	St	Activity	Obj coef Marginal	Lower bound Upper bound	Activity range	Obj coef range	Obj value at break point	Limiting variable	
1	x1	BS	20.00000	3.00000 . .	. +Inf	-Inf 40.00000	2.00000 4.00000	160.00000 200.00000	x3 x4	
2	x2	BS	60.00000	2.00000 . .	. +Inf	20.00000 80.00000	1.50000 3.00000	150.00000 240.00000	x4 x3	

Fig. 1. – Rapport GLPK - Analyse de sensibilité de la fonction objectif

Colonnes du rapport :

- **Column name** : Nom de la variable de décision.
- **St (Status)** : Statut de la variable (BS = Basic, dans la base ; NU = Non-Upper, à sa borne supérieure).
- **Activity** : Valeur de la variable à l'optimum (solution actuelle).
- **Obj coef** : Coefficient actuel de la variable dans la fonction objectif.
- **Activity range** : Intervalle de variation de la valeur de la variable (si elle reste dans la base).
- **Obj coef range** : Intervalle de stabilité du coefficient — tant que le coefficient reste dans cet intervalle, la solution optimale ne change pas.
- **Obj value at break point** : Valeur de la fonction objectif aux bornes de l'intervalle de stabilité.
- **Limiting variable** : Variable qui entre/sort de la base aux bornes de l'intervalle.

Interprétation pour x_1 :

- **Activity** = 20 : À l'optimum actuel, on produit 20 soldats.
- **Obj coef** = 3 : Le profit unitaire actuel est de 3 francs.
- **Obj coef range** = [2, 4] : Tant que le profit unitaire des soldats varie entre 2 et 4 francs, la solution optimale reste $(x_1, x_2) = (20, 60)$.
- **Obj value at break point** = 160 / 200 :
 - Si $c_1 = 2$, le profit total devient 160 francs.
 - Si $c_1 = 4$, le profit total devient 200 francs.
- **Limiting variable** = x3 / x4 : Aux bornes, les contraintes associées à x_3 et x_4 deviennent critiques.
- **Activity range** = $[-\infty, 40]$: Si le coefficient sort de l'intervalle, x_1 pourrait varier entre 0 et 40.

4.2 Sensibilité du second membre

Reprendons le problème de Geppetto SA et modifions le second membre b_2 de la deuxième contrainte (actuellement 80).

4.2.1 Effet géométrique

Modifier b_2 provoque un **déplacement parallèle** de la droite délimitant le demi-plan admissible associé à cette contrainte.

La région admissible se déforme, et le sommet optimal peut changer ou se déplacer le long d'une arête.

4.2.2 Analyse similaire

De manière analogue à l'analyse de la fonction objectif, on peut déterminer :

- Un **intervalle de variation** $[b_{\min}, b_{\max}]$ pour le second membre.
- Dans cet intervalle, la **base optimale** reste la même (mais les valeurs des variables changent).
- La valeur de la fonction objectif varie linéairement avec le second membre.
- Aux bornes de l'intervalle, la base optimale change.

4.2.3 Rapport de sensibilité dans GLPK

La première page du rapport contient l'analyse pour les contraintes (second membre) :

GLPK 4.65 - SENSITIVITY ANALYSIS REPORT								Page 1
Problem:	geppetto							
Objective:	$z = 180$ (MAXimum)							
No.	Row name	St	Activity	Slack Marginal	Lower bound Upper bound	Activity range	Obj coef range	Obj value at break point Limiting variable
1	z	BS	180.00000	-180.00000	. -Inf +Inf	160.00000 180.00000	-1.00000 +Inf	. +Inf x4
2	x3	NU	100.00000	. 1.00000	-Inf 100.00000	80.00000 120.00000	-1.00000 +Inf	160.00000 x1 200.00000 x5
3	x4	NU	80.00000	. 1.00000	-Inf 80.00000	60.00000 100.00000	-1.00000 +Inf	160.00000 x5 200.00000 x1
4	x5	BS	20.00000	20.00000	-Inf 40.00000	50.00000	-1.00000 1.00000	160.00000 x3 200.00000 x4

Fig. 2. – Rapport GLPK - Analyse de sensibilité des contraintes

Colonnes du rapport :

- **Row name** : Nom de la contrainte (ligne du problème).
- **St (Status)** : Statut de la contrainte (BS = Basic, contrainte non saturée ; NU = Non-Upper, contrainte saturée à sa borne).
- **Activity** : Valeur du membre de gauche de la contrainte à l'optimum.
- **Slack Marginal** : Marge (slack) et **prix dual** (ou variable duale) de la contrainte.
- **Lower/Upper bound** : Bornes actuelles du second membre.
- **Activity range** : **Intervalle de stabilité** du second membre — tant qu'il reste dans cet intervalle, la base optimale ne change pas.
- **Obj coef range** : Intervalle de variation du prix dual.
- **Obj value at break point** : Valeur de la fonction objectif aux bornes de l'intervalle.
- **Limiting variable** : Variable qui entre/sort de la base aux bornes.

Interprétation pour la contrainte x3 (finissage : $x_1 + x_2 \leq 100$) :

- **Activity** = 100 : La contrainte est **saturée** (on utilise les 100 heures disponibles).
- **Slack** = 0 (implicite par NU) : Il n'y a pas de marge, la contrainte est active.
- **Marginal** = 1 : Le **prix dual** est 1. Cela signifie que si on augmente b_1 d'une unité (de 100 à 101), le profit optimal augmente de 1 franc.
- **Activity range** = [80, 120] : Tant que le second membre b_1 varie entre 80 et 120 heures, la base optimale reste la même (mais les valeurs des variables changent).

- **Obj value at break point** = 160 / 200 :
 - ▶ Si $b_1 = 80$, le profit devient 160 francs.
 - ▶ Si $b_1 = 120$, le profit devient 200 francs.

Interprétation pour la contrainte x_5 (demande soldats : $x_1 \leq 40$) :

- **Activity** = 20 : On produit 20 soldats (bien en-dessous de la limite de 40).
- **Slack** = 20 : Il reste une marge de 20 soldats avant d'atteindre la limite.
- **Marginal** = 0 : Le prix dual est nul. Augmenter cette limite n'améliorerait pas le profit (la contrainte n'est pas active).
- **Activity range** = $[., 50]$: Si la limite passe au-dessus de 50, la base optimale change.

Principe du prix dual : Le prix dual d'une contrainte indique de combien augmente (ou diminue) la valeur optimale si on augmente le second membre de cette contrainte d'une unité, tant qu'on reste dans l'intervalle de stabilité. Les contraintes avec un prix dual élevé sont les plus « précieuses » (goulots d'étranglement).

4.3 Utilité de l'analyse de sensibilité

L'analyse de sensibilité permet de :

- **Évaluer la robustesse** de la solution face aux imprécisions des données.
- **Identifier les paramètres critiques** dont la variation impacte fortement la solution.
- **Répondre à des questions de type « what-if »** sans résoudre à nouveau le problème.
- **Négocier les ressources** en connaissant leur impact marginal (prix duaux).

5 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.

5.1 Programme linéaire binaire

Un programme linéaire binaire (PLB) est un type particulier de PLNE où toutes les variables ne peuvent prendre que deux valeurs : 0 ou 1. Ces variables binaires sont utilisées pour modéliser des **décisions de type oui/non**:

- Sélectionner ou non un projet
- Construire ou non une usine
- Inclure ou non un élément dans un ensemble
- Activer ou non une ressource

Les PLB sont particulièrement adaptés pour résoudre des problèmes de :

- **Sélection optimale** (choisir les meilleurs éléments parmi un ensemble)
- **Affectation** (assigner des ressources à des tâches)
- **Couverture** (couvrir un ensemble d'exigences avec un minimum de ressources)
- **Ordonnancement** (planifier des activités dans le temps)

5.1.1 Exemple : Sélection de projets

Une entreprise dispose d'un budget de 100'000 francs et doit choisir parmi 4 projets à financer. Chaque projet a un coût et un bénéfice estimé :

Projet	Coût (kCHF)	Bénéfice
1	40	60
2	30	45
3	50	70
4	35	50

On introduit une variable binaire x_i pour chaque projet :

$$x_i = \begin{cases} 1 & \text{si le projet } i \text{ est sélectionné} \\ 0 & \text{sinon} \end{cases} \quad i = 1, 2, 3, 4$$

Le problème se formule ainsi :

$$\begin{aligned} \text{Max } z &= 60x_1 + 45x_2 + 70x_3 + 50x_4 \\ \text{s.c. } &40x_1 + 30x_2 + 50x_3 + 35x_4 \leq 100 \quad (\text{budget}) \\ &x_i \in \{0, 1\} \quad i = 1, 2, 3, 4 \end{aligned}$$

Solution optimale :

$$x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1 \quad \text{avec } z = 155$$

5.2 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.

5.2.1 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 modélise de la manière suivante :

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

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

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

5.2.2 Fonction objectif avec coûts fixes

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

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

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).

5.2.3 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

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

Nous pouvons alors modéliser x de la manière suivante :

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

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 $x = a$.

5.2.4 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 :

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

5.2.5 Valeurs absolues

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

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

Exemple

$$|2x_1 + 3x_2| \leq 12 \iff \begin{cases} 2x_1 + 3x_2 \leq 12 \\ 2x_1 + 3x_2 \geq -12 \end{cases}$$

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

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

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 :

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

5.2.5.1 Exemple

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

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

6 Résolution des programmes linéaires en nombres entiers

6.1 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 :

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

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

6.2 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.

6.3 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.

6.4 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) :

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

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-optimale).

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