1

Algorithme du Simplexe

## 1.1 Forme algébrique de l'algorithme

- **Forme standard** :

$$\text{Minimiser } z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$
$$\text{s.a.} \begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leqslant b_1 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leqslant b_m \\ x_1, x_2, \ldots, x_n \geqslant 0 \end{cases}$$

- **Introduction des variables d'écart** $s_i$ :

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + s_i = b_i, \quad s_i \geqslant 0$$

- **Fonction objectif ajustée** :

$$z + c_1x_1 + c_2x_2 + \cdots + c_nx_n = 0$$

- **Choix de la variable entrante** :
  - Sélectionner la variable avec le coefficient **le plus négatif** dans la fonction objectif.
- **Détermination de la variable sortante** :
  - Calculer le **ratio** pour chaque contrainte :

$$\boxed{\text{Ratio} = \frac{b_i}{a_{ij}}} \quad \text{avec } a_{ij} > 0$$

  - La variable avec le plus petit ratio positif est la **variable sortante**.
  - Si tous les $a_{ij}$ sont négatifs, on conclue alors que le problème est **non borné** .
- **Pivot** :
  - Mettre à jour le tableau en pivotant sur le coefficient correspondant.
- **Critère d'optimalité** :
  - Si tous les coefficients $\overline{c}_j$de la fonction objectif sont tels que $\overline{c}_j \geqslant 0$, la solution est optimale.
- **Formules importantes** :
  - **Coefficient réduit** :

$$\boxed{\overline{c}_j = c_j - \sum_i c_i^B a_{ij}}$$

  - **Mise à jour des coefficients** :

Nouvelle ligne pivot :

$$\boxed{\text{Ligne pivot} \div \text{Coefficient pivot}}$$

Autres lignes :

$$\boxed{\text{Ligne} - (\text{Coefficient} \times \text{Nouvelle ligne pivot})}$$

- **Résumé des étapes** :
  - Formuler le problème en forme standard.
  - Identifier la variable entrante (coefficient le plus négatif).
  - Trouver la variable sortante (plus petit ratio positif).
  - Effectuer le pivot.
  - Répéter jusqu'à optimalité.

## 1.2 Forme Tabulée de l'algorithme

- **Construction du Tableau Simplexe** :
  - Organiser les équations du problème en **forme standard** dans un tableau.
  - La première colonne contient les **variables de base** (variables dépendantes).
  - Chaque ligne représente une équation, y compris la fonction objectif négative $(-z)$.
- **Choix de la Variable Entrante** :
  - Sélectionner la variable avec le coefficient le plus négatif dans la ligne $(-z)$.
  - Cette variable améliorera le plus la valeur de l'objectif.
- **Détermination de la Variable Sortante** :
  - Calculer les **ratios** pour chaque variable de base :

$$\boxed{\text{Ratio} = \frac{\text{Terme de droite}}{\text{Coefficient de la variable entrante}}}$$

  - Ignorer les coefficients négatifs ou nuls de la variable entrante.
  - La variable avec le plus petit ratio positif est la **variable sortante**.
- **Pivot** :
  - Identifier le **coefficient pivot** à l'intersection de la colonne de la variable entrante et de la ligne de la variable sortante.
  - Diviser la ligne pivot par le coefficient pivot pour obtenir un 1.
  - Utiliser des opérations sur les lignes pour annuler les autres coefficients dans la colonne de la variable entrante.
- **Itération** :
  - Répéter le processus de sélection des variables entrante et sortante, suivi du pivot, jusqu'à ce que tous les coefficients de $(-z)$ soient positifs ou nuls.
- **Solution Optimale** :

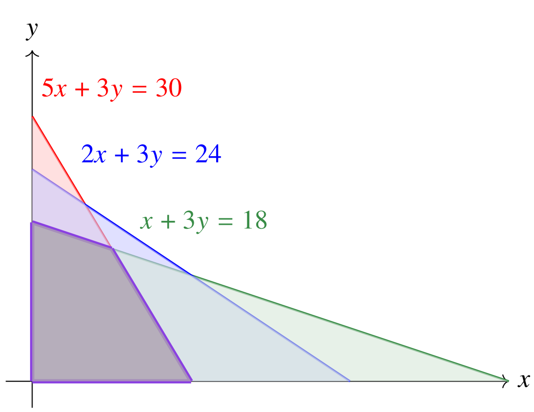
- ▷ Lorsque tous les coefficients de $(-z)$ sont positifs ou nuls, la solution optimale est atteinte.
  - ▷ Les valeurs des variables de base se trouvent dans la colonne des termes de droite.
- **Nombre de Solutions de Base** :
  - ▷ Pour un système avec $n$ variables et $m$ équations :

$$\boxed{\text{Nombre de solutions de base} = \binom{n}{m} = \frac{n!}{m!(n-m)!}}$$

- **Résumé des Étapes** :
  - ▷ **Organiser** les équations en tableau.
  - ▷ **Choisir** la variable entrante $(x_s)$ avec le coefficient le plus négatif dans $(-z)$.
  - ▷ **Déterminer** la variable sortante $(x_r)$ avec le plus petit ratio positif.
  - ▷ **Effectuer** le pivot sur le coefficient pivot.
  - ▷ **Répéter** jusqu'à optimalité.

## 1.3 Résolution Graphique

- **Méthode de Résolution Graphique** (pour problèmes à deux variables) :
  - ▷ **Tracer les contraintes** sur un plan cartésien :
    - ▷ Convertir chaque inégalité en équation pour tracer la droite correspondante.
    - ▷ Identifier la zone admissible (région des solutions réalisables) en tenant compte des inégalités.
  - ▷ **Déterminer les sommets** de la zone admissible :
    - ▷ Trouver les points d'intersection des droites (sommets du polygone formé).
  - ▷ **Calculer la valeur de la fonction objectif** en chaque sommet :
    - ▷ Évaluer $z = c_1 x + c_2 y$ pour chaque couple $(x, y)$.
  - ▷ **Choisir la solution optimale** :
    - ▷ Pour une maximisation, sélectionner le sommet avec la valeur de $z$ la plus élevée.
    - ▷ Pour une minimisation, choisir le sommet avec la valeur de $z$ la plus faible.




### Notes Importantes

- **Dans un problème à deux variables**, la zone admissible est un polygone convexe sur le plan $xy$, et la solution optimale se trouve toujours à l'un des sommets de ce polygone.
- **Vérification de la Faisabilité** :
  - ▷ S'assurer que les points trouvés respectent toutes les contraintes.
- **Cas Particuliers** :
  - ▷ **Solutions Multiples** : Si la fonction objectif est parallèle à une contrainte bordant la région admissible.
  - ▷ **Région Non Bornée** : Si la région admissible n'est pas limitée, la solution optimale peut être non bornée.

## 1.4 Forme Générale de l'Algorithme

- **Notations** :
  - ▷ $x_j$ : Variables de décision, $j = 1, 2, \ldots, n$.
  - ▷ $c_j$ : Coefficients dans la fonction objectif.
  - ▷ $a_{ij}$ : Coefficients des contraintes, $i = 1, 2, \ldots, m$.
  - ▷ $b_i$ : Termes de droite des contraintes.
- **Forme Standard** :

$$\text{Minimiser } z = \sum_{j=1}^{n} c_j x_j$$

$$\text{sous contraintes } \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \ldots, m$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n$$

- **Tableau Simplexe Général** :
  - ▷ Les variables de base $x_i$ forment une base identité.
  - ▷ Les variables non de base sont initialement fixées à zéro.
- **Choix de la Variable Entrante** :
  - ▷ Si tous les coefficients réduits $\overline{c}_j \geqslant 0$, la solution est optimale.
  - ▷ Sinon, choisir $x_s$ tel que :

$$\boxed{\overline{c}_s = \min\{c_j \mid j = 1, 2, \ldots, n\}}$$

- **Détermination de la Variable Sortante** :
  - ▷ Calculer les ratios pour $\overline{a}_{is} > 0$ :

$$\boxed{\text{Ratio} = \frac{\overline{b}_i}{\overline{a}_{is}}}$$

▹ La variable sortante $x_r$ correspond au plus petit ratio positif.
▹ Si $\overline{a}_{is} \leqslant 0$ pour tout $i$, le problème est non borné.

- **Pivot et Mise à Jour du Tableau** :
  - ▹ **Ligne Pivot** $(r)$ :

$$\boxed{\text{Nouvelle ligne } r = \frac{\text{Ancienne ligne } r}{\overline{a}_{rs}}}$$

  - ▹ **Autres Lignes** $(i \neq r)$ :

$$\boxed{\text{Nouvelle ligne } i = \text{Ancienne ligne } i - \overline{a}_{is} \times \text{Nouvelle ligne } r}$$

  - ▹ **Fonction Objectif** :

$$\boxed{\overline{\overline{c}}_j = \overline{c}_j - \overline{c}_s \times \overline{a}_{rj}}$$

$$\boxed{\overline{\overline{z}} = \overline{z} + \overline{c}_s \times \overline{b}_r}$$

- **Résumé des Étapes de l'Algorithme** :
  - ▹ **Initialisation** : Formuler le problème en forme standard et construire le tableau initial.
  - ▹ **Itération** :
    - ▹ Choisir la variable entrante $x_s$ avec le $\overline{c}_s$ le plus négatif.
    - ▹ Déterminer la variable sortante $x_r$ en calculant les ratios.
    - ▹ Effectuer le pivot pour mettre à jour le tableau.
  - ▹ **Vérification d'Optimalité** :
    - ▹ Si tous les $\overline{c}_j \geqslant 0$, la solution optimale est atteinte.
    - ▹ Sinon, répéter l'itération.
- **Formules Clés** :
  - ▹ **Mise à Jour des Coefficients** :

$$\boxed{\tilde{\overline{a}}_{ij} = \overline{a}_{ij} - \overline{a}_{is} \times \overline{a}_{rj}}$$

  - ▹ **Mise à Jour des Termes de Droite** :

$$\boxed{\tilde{\overline{b}}_i = \overline{b}_i - \overline{a}_{is} \times \overline{b}_r}$$

## 1.5 Forme Matricielle de l'Algorithme

### Notations Matricielles

- **Variables de Décision** :

$$\mathbf{x}_{n\times 1} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- **Coefficients de l'Objectif** :

$$\mathbf{c}_{n\times 1} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

- **Matrice des Contraintes** :

$$\mathbf{A}_{m\times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

- **Termes de Droite** :

$$\mathbf{b}_{m\times 1} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

### Formulation Matricielle du Problème

Le problème de programmation linéaire en forme standard peut être écrit en notation matricielle comme suit :

$$\begin{aligned} \text{Minimiser} \quad & z = \mathbf{c}^T\mathbf{x} \\ \text{sous contraintes} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geqslant 0 \end{aligned}$$

### Interprétation

▹ **Objectif** : $\mathbf{c}^T\mathbf{x} = \sum_{j=1}^n c_j x_j$
▹ **Contraintes** : Chaque équation $\sum_{j=1}^n a_{ij}x_j = b_i$ est représentée par le produit matriciel $\mathbf{A}\mathbf{x} = \mathbf{b}$
▹ **Non-négativité** : $\mathbf{x} \geqslant 0$ signifie que chaque $x_j \geqslant 0$

### Exemple : Problème du Restaurateur

Considérons le problème du restaurateur avec les variables $x$ et $y$, et les variables d'écart $u, p, h$.

$$\begin{aligned} &\text{Minimiser} \quad z = 8x + 6y \\ &\text{sous contraintes} \\ &5x + 3y + u = 30 \\ &2x + 3y + p = 24 \\ &x + 3y + h = 18 \\ &x, y, u, p, h \geqslant 0 \end{aligned}$$

En notation matricielle :

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ u \\ p \\ h \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} -8 \\ -6 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 5 & 3 & 1 & 0 & 0 \\ 2 & 3 & 0 & 1 & 0 \\ 1 & 3 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 30 \\ 24 \\ 18 \end{bmatrix}$$

## Base et Variables de Base

- Une **base** $B$ est une sous-matrice $m \times m$ non singulière de $\mathbf{A}$.
- Les variables correspondant aux colonnes de $B$ sont les **variables de base**.
- Les autres variables sont les **variables hors-base**.

### Exemple de Bases

$$B_1 = \begin{bmatrix} u & p & h \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} x & u & h \\ 5 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad B_3 = \begin{bmatrix} x & p & y \\ 5 & 0 & 3 \\ 2 & 1 & 3 \\ 1 & 0 & 3 \end{bmatrix}$$

## Partitionnement des Matrices

- **Variables** :
$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_R \end{bmatrix}$$
où $\mathbf{x}_B$ sont les variables de base et $\mathbf{x}_R$ les variables hors-base.
- **Coefficients de l'Objectif** :
$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_B \\ \mathbf{c}_R \end{bmatrix}$$
- **Matrice des Contraintes** :
$$\mathbf{A} = [B \ : \ R]$$
où $B$ est la base et $R$ le reste de la matrice.

## Formulation Réécrite

Le problème peut être réécrit comme :

$$\begin{aligned} &\text{Minimiser} \quad z = \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_R^T \mathbf{x}_R \\ &\text{sous contraintes} \quad B\mathbf{x}_B + R\mathbf{x}_R = \mathbf{b} \\ &\mathbf{x}_B \geqslant 0, \quad \mathbf{x}_R \geqslant 0 \end{aligned}$$

## Résolution par le Simplexe en Notation Matricielle

1. **Expression de $\mathbf{x}_B$** :
$$\mathbf{x}_B = B^{-1}\mathbf{b} - B^{-1}R\mathbf{x}_R$$
2. **Substitution dans l'Objectif** :
$$z = \mathbf{c}_B^T(B^{-1}\mathbf{b} - B^{-1}R\mathbf{x}_R) + \mathbf{c}_R^T \mathbf{x}_R$$
3. **Définition des Multiplicateurs du Simplexe** :
$$\pi^T = \mathbf{c}_B^T B^{-1}$$
4. **Calcul des Coûts Réduits** :
$$\tilde{\mathbf{c}}_R^T = \mathbf{c}_R^T - \pi^T R$$
5. **Nouvelle Forme de l'Objectif** :
$$z = \pi^T \mathbf{b} + \tilde{\mathbf{c}}_R^T \mathbf{x}_R$$

## Exemple Numérique

- **Matrice de Base** $B$ et son inverse $B^{-1}$
$$B = \begin{bmatrix} 5 & 0 & 3 \\ 2 & 1 & 3 \\ 1 & 0 & 3 \end{bmatrix}, \quad B^{-1} = \begin{bmatrix} 1/4 & 0 & -1/4 \\ -1/4 & 1 & -3/4 \\ -1/12 & 0 & 5/12 \end{bmatrix}$$
- **Calcul de $\pi^T$** :
$$\pi^T = \mathbf{c}_B^T B^{-1} = \begin{bmatrix} -8 & 0 & -6 \end{bmatrix} \begin{bmatrix} 1/4 & 0 & -1/4 \\ -1/4 & 1 & -3/4 \\ -1/12 & 0 & 5/12 \end{bmatrix} = \begin{bmatrix} -\frac{3}{2} & 0 & -\frac{1}{2} \end{bmatrix}$$
- **Calcul des Coûts Réduits** :
$$\tilde{\mathbf{c}}_R^T = \mathbf{c}_R^T - \pi^T R = \begin{bmatrix} 0 & 0 \end{bmatrix} - \begin{bmatrix} -\frac{3}{2} & 0 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & \frac{1}{2} \end{bmatrix}$$
- **Valeur de l'Objectif** :
$$z = \pi^T \mathbf{b} = \begin{bmatrix} -\frac{3}{2} & 0 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} 30 \\ 24 \\ 18 \end{bmatrix} = -54$$

## Résumé des Étapes

1. **Initialisation** : Choisir une base initiale $B$ (par exemple, les variables d'écart).
2. **Calculer** $B^{-1}$ et $\pi^T = \mathbf{c}_B^T B^{-1}$.
3. **Calculer** les coûts réduits $\tilde{\mathbf{c}}_R^T = \mathbf{c}_R^T - \pi^T R$.
4. **Vérifier** la condition d'optimalité : si tous les $\tilde{c}_j \geqslant 0$, la solution est optimale.
5. **Sinon**, choisir la variable entrante correspondante au $\tilde{c}_j < 0$ le plus négatif.
6. **Déterminer** la variable sortante en utilisant la règle du rapport minimal.
7. **Mettre à jour** la base $B$ et répéter les étapes jusqu'à l'optimalité.

## 1.6 Analyse Post-Optimale

- **Modification des Termes de Droite ($b$)** :
  - Si les termes de droite $b$ sont modifiés par $\Delta b$, la nouvelle solution est :

$$\boxed{\overline{\overline{b}} = \overline{b} + B^{-1}\Delta b}$$

  - La nouvelle valeur de la fonction objectif est :

$$\boxed{\tilde{z} = \overline{z} + \pi^T \Delta b}$$

  - **Interprétation** :
    - Si $\overline{\overline{b}} \geqslant 0$, la base reste optimale.
    - Si $\overline{\overline{b}}$ contient des valeurs négatives, une réoptimisation est nécessaire.
- **Modification d'un Coût dans la Fonction Objectif ($c_j$)** :
  - Si un coût $c_j$ est modifié par $\Delta c_j$, le nouveau coefficient réduit est :

$$\boxed{\overline{\tilde{c}}_j = \overline{c}_j + \Delta c_j}$$

  - **Interprétation** :
    - Si $\overline{\tilde{c}}_j \geqslant 0$ pour toutes les variables hors-base, la solution reste optimale.
    - Sinon, il faut réoptimiser en introduisant la variable correspondante dans la base.
- **Modification d'une Variable Hors-Base (Colonne $a_j$)** :
  - Si une colonne $a_j$ est modifiée par $\Delta a_j$, le nouveau coefficient réduit devient :

$$\boxed{\overline{\tilde{c}}_j = \overline{c}_j - \pi^T \Delta a_j}$$

  - **Interprétation** :
    - Si $\overline{\tilde{c}}_j \geqslant 0$, la base reste optimale.
    - Si $\overline{\tilde{c}}_j < 0$, une réoptimisation est nécessaire.
- **Ajout d'une Nouvelle Variable (Nouvelle Action)** :
  - Pour une nouvelle variable $x_r$ avec coût $c_r$ et colonne $a_r$ :

$$\boxed{\tilde{a}_r = B^{-1} a_r}$$

$$\boxed{\tilde{c}_r = c_r - \pi^T a_r}$$

  - **Interprétation** :
    - Si $\tilde{c}_r \geqslant 0$, la nouvelle variable n'améliore pas la solution actuelle.
    - Si $\tilde{c}_r < 0$, il est avantageux d'introduire $x_r$ dans la base et de réoptimiser.
- **Généralités sur l'Analyse Post-Optimale** :
  - L'analyse post-optimale permet d'évaluer l'impact de modifications mineures sans refaire tout le calcul.
  - Elle utilise les informations du dernier tableau optimal, notamment $B^{-1}$ et les multiplicateurs de Lagrange $\pi$.
  - Elle est utile pour les **analyses de sensibilité** et la prise de décision informée.

2

## Programmation Linéaire en Nombres Entiers

- **Introduction** :
  - La programmation linéaire en nombres entiers (PLNE) est utilisée lorsque les variables doivent prendre des valeurs entières.
  - Les solutions fractionnaires ne sont pas acceptables dans certains contextes (exemple : on ne peut pas commander une fraction d'un produit).
- **Modèles Binaires** :
  - Utilisation de variables binaires $x_i \in \{0, 1\}$ pour modéliser des décisions oui/non.
  - Exemples d'applications : sélection de projets, affectation de tâches, problèmes de routage.
- **Problème de Coloration de Graphe** :
  - Objectif : Colorier les sommets d'un graphe de sorte que deux sommets adjacents n'aient pas la même couleur, en minimisant le nombre de couleurs utilisées.
  - **Variables** :

$$c_i = \begin{cases} 1 & \text{si la couleur } i \text{ est utilisée} \\ 0 & \text{sinon} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{si le sommet } j \text{ est colorié avec la couleur } i \\ 0 & \text{sinon} \end{cases}$$

  - **Contraintes** :
    - Chaque sommet doit être colorié avec exactement une couleur :

$$\sum_{i=1}^{m} x_{ij} = 1, \quad \forall j \in V$$

    - Deux sommets adjacents ne peuvent pas avoir la même couleur :

$$x_{ij}+x_{ik} \leqslant c_i, \quad \forall (j,k) \in E, \quad \forall i = 1, \dots, m$$

    - Liaison entre $x_{ij}$ et $c_i$ :

$$x_{ij} \leqslant c_i, \quad \forall i = 1, \dots, m, \quad \forall j \in V$$

  - **Objectif** :

$$\text{Minimiser} \quad \sum_{i=1}^{m} c_i$$

- **Principes Généraux en PLNE** :
  - **Relaxation Continue** :
    - En relâchant les contraintes d'intégralité, on obtient une borne inférieure sur la valeur optimale.
    - Le domaine faisable de la PLNE est inclus dans celui de sa relaxation :

$$F(P) \subseteq F(\tilde{P}), \quad v(\tilde{P}) \leqslant v(P)$$

  - **Optimalité des Solutions Entières** :
    - Si la solution optimale de la relaxation continue est entière, elle est aussi optimale pour le problème entier.
- **Exemple Illustratif** :
  - Considérons le problème :

$$\begin{aligned} \text{Minimiser} \quad & z = -x_1 - 5x_2 \\ \text{sous contraintes} \quad & x_1 + 10x_2 \leqslant 20 \\ & x_1 \leqslant 2 \\ & x_1, x_2 \geqslant 0 \text{ et entiers} \end{aligned}$$

  - La relaxation continue permet de trouver une solution fractionnaire qui sert de borne inférieure.
  - Les solutions obtenues par arrondi ou troncature ne sont pas nécessairement optimales pour le problème entier.

# Algorithme Branch-and-Bound

- **Introduction** :
  - Méthode pour résoudre les problèmes de PLNE en explorant systématiquement les sous-problèmes.
  - Combine des techniques de séparation (branching) et d'évaluation (bounding).
- **Branchement (Branching)** :
  - Si la solution optimale de la relaxation est fractionnaire, on crée deux sous-problèmes en imposant des contraintes supplémentaires sur une variable fractionnaire $x_j$ :

$$P_1 : x_j \leqslant \lfloor x_j^* \rfloor$$
$$P_2 : x_j \geqslant \lceil x_j^* \rceil$$

- **Évaluation des Sous-Problèmes (Bounding)** :
  - Résolution de la relaxation continue de chaque sous-problème pour obtenir une borne inférieure.
  - Mise à jour de la borne supérieure $\overline{z}$ lorsque des solutions entières meilleures sont trouvées.
  - **Relation des Bornes** :

$$\overline{z} \geqslant z^* \geqslant z, \quad \forall \text{ sous-problèmes}$$

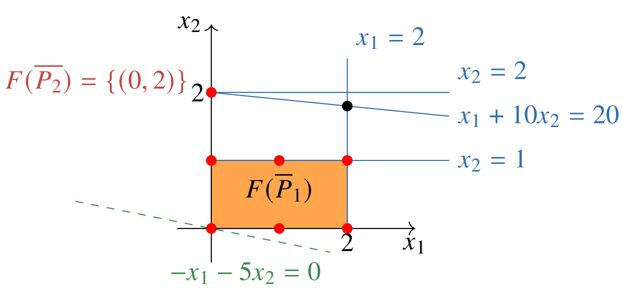
- **Critères d'Arrêt** :
  - Un sous-problème peut être éliminé si :
    - Son domaine faisable est vide.

- ▷ La solution optimale est entière et son coût est supérieur ou égal à $\overline{z}$.
- ▷ La borne inférieure du sous-problème est supérieure ou égale à $\overline{z}$.

- ► **Algorithme (Pseudo-Code)** :
  1. **Initialisation** : Résoudre la relaxation continue du problème initial pour obtenir une borne inférieure $\underline{z}$.
  2. **Mise à Jour** :
     - ▷ Si la solution est entière, elle devient la borne supérieure $\overline{z}$.
     - ▷ Sinon, on branche sur une variable fractionnaire pour créer des sous-problèmes.
  3. **Exploration** :
     - ▷ Résoudre les relaxations continues des sous-problèmes.
     - ▷ Appliquer les critères d'arrêt pour éliminer des branches.
     - ▷ Mettre à jour $\overline{z}$ si une meilleure solution entière est trouvée.
  4. **Itération** : Répéter l'exploration jusqu'à ce que toutes les branches soient explorées ou éliminées.
- ► **Exemple Illustratif** :
  - ▷ Considérons le problème :

$$\begin{array}{rl} \text{Minimiser} & z = -x_1 - 5x_2 \\ \text{sous contraintes} & x_1 + 10x_2 \leqslant 20 \\ & x_1 \leqslant 2 \\ & x_1, x_2 \geqslant 0 \text{ et entiers} \end{array}$$

  - ▷ **Étape 1** : Résoudre la relaxation continue pour obtenir $x_1^* = 2$, $x_2^* = 1.8$, $z^* = -11$.
  - ▷ **Branchement sur** $x_2$ :
    - ▷ **Sous-problème** $P_1$ : $x_2 \leqslant 1$.
    - ▷ **Sous-problème** $P_2$ : $x_2 \geqslant 2$.
  - ▷ **Résolution de** $P_1$ :
    - ▷ Solution entière : $x_1 = 2$, $x_2 = 1$, $z = -7$.
    - ▷ Mise à jour de $\overline{z} = -7$.
  - ▷ **Résolution de** $P_2$ :
    - ▷ Solution entière : $x_1 = 0$, $x_2 = 2$, $z = -10$.
    - ▷ Mise à jour de $\overline{z} = -10$ car meilleure que la précédente.
  - ▷ **Conclusion** : La solution optimale est $x_1 = 0$, $x_2 = 2$, avec $z = -10$.



3

## Optimisation de Graphes et Réseaux

### Notions de Base

- **Graphe Non Orienté** :
  - Un graphe $G = (V, E)$ où $V$ est l'ensemble des sommets et $E$ l'ensemble des arêtes.
  - Une arête $(i, j)$ relie les sommets $i$ et $j$.
  - Deux sommets sont **adjacents** s'ils sont reliés par une arête.
  - Une arête est **incidente** à ses sommets.
- **Chaînes et Cycles** :
  - Une **chaîne** est une suite d'arêtes connectant une séquence de sommets.
  - Un **cycle** est une chaîne où le premier et le dernier sommet sont identiques.
- **Graphe Connexe** :
  - Un graphe est connexe s'il existe une chaîne entre chaque paire de sommets.

### Graphe Orienté et Concepts Associés

- **Graphe Orienté** :
  - Un graphe $G = (V, A)$ où $A$ est l'ensemble des arcs (arêtes orientées).
  - Un arc $(i, j)$ va du sommet $i$ au sommet $j$.
- **Chemins et Circuits** :
  - Un **chemin** est une suite d'arcs orientés dans la même direction.
  - Un **circuit** est un chemin fermé (départ et arrivée au même sommet).

### Concepts de Flot et Contraintes sur les Arcs

- **Flot sur les Arcs** :
  - Chaque arc $(i, j)$ a :
    - Une **borne inférieure** $l_{ij}$ (minimum de flot).
    - Une **capacité** $u_{ij}$ (maximum de flot).
    - Un **coût** $c_{ij}$ par unité de flot.
- **Objectif** :
  - Maximiser le flot total du **source** $s$ au **puits** $t$.
  - Minimiser le coût total associé au flot.

## Problèmes de Flots à Coût Minimal

### Notations de Base

- $G = (V, A)$ : Graphe orienté avec sommets $V$ et arcs $A$.
- $c_{ij}$ : Coût par unité de flot sur l'arc $(i, j)$.
- $x_{ij}$ : Flot sur l'arc $(i, j)$.
- $u_{ij}$ : Capacité maximale de l'arc $(i, j)$.
- $s$ : Source du réseau.
- $t$ : Puits du réseau.

### Formulation Mathématique

$$\text{Minimiser} \quad z = \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$\text{sous contraintes} \quad \sum_{j\in P_i} x_{ij} - \sum_{j\in B_i} x_{ji} = \begin{cases} d & \text{si } i = s \\ -d & \text{si } i = t \\ 0 & \text{sinon} \end{cases}$$

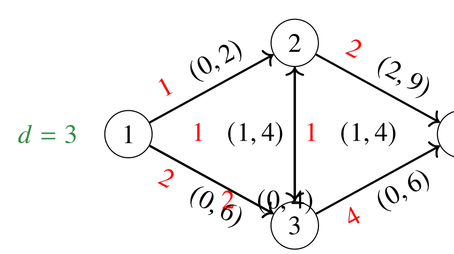
$$l_{ij} \leqslant x_{ij} \leqslant u_{ij}, \quad \forall (i, j) \in A$$

- $P_i$ : Ensemble des sommets atteints depuis $i$ (arcs sortants).
- $B_i$ : Ensemble des sommets rejoignant $i$ (arcs entrants).
- $d$ : Quantité de flot à acheminer de $s$ vers $t$.

### Interprétation

- **Conservation du Flot** : Le flot entrant moins le flot sortant est égal à la demande ou l'offre en chaque sommet.
- **Respect des Capacités** : Le flot sur chaque arc doit respecter ses bornes.
- **Objectif** : Minimiser le coût total du transport du flot.

## Exemple de Problème à Coût Minimal




- **Réseau** :
  - Sommets : 1 (source), 2, 3, 4 (puits).
  - Arcs avec coûts $c_{ij}$, capacités $u_{ij}$ et bornes inférieures $l_{ij}$.
- **Objectif** :

$$\text{Minimiser} \quad z = x_{12} + 2x_{13} + 2x_{23} + 2x_{24} + x_{32} + 4x_{34}$$

- **Contraintes de Conservation du Flot** :

$$\begin{cases} x_{12} + x_{13} = 3 & \text{(sommet 1)} \\ -x_{12} + x_{23} + x_{24} - x_{32} = 0 & \text{(sommet 2)} \\ -x_{13} - x_{23} + x_{32} + x_{34} = 0 & \text{(sommet 3)} \\ -x_{24} - x_{34} = -3 & \text{(sommet 4)} \end{cases}$$

- **Contraintes de Capacité et Bornes** :

$$0 \leqslant x_{12} \leqslant 2, \quad 0 \leqslant x_{13} \leqslant 4, \quad 0 \leqslant x_{23} \leqslant 6,$$
$$2 \leqslant x_{24} \leqslant 9, \quad 1 \leqslant x_{32} \leqslant 4, \quad 0 \leqslant x_{34} \leqslant 6$$

# Problème de Flot avec Plusieurs Sources et Puits

## Formulation Générale

- **Objectif** :

$$\text{Minimiser} \quad z = \sum_{(i,j)\in A} c_{ij} x_{ij}$$

- **Contraintes de Conservation du Flot** :

$$\sum_{j\in P_i} x_{ij} - \sum_{j\in B_i} x_{ji} = \begin{cases} d_i & \text{si } i \in S \text{ (sources)} \\ -d_j & \text{si } i \in T \text{ (puits)} \\ 0 & \text{sinon} \end{cases}$$

- **Contraintes de Capacité** :

$$0 \leqslant x_{ij} \leqslant u_{ij}, \quad \forall (i,j) \in A$$

## Interprétation

- Chaque source $s_i$ fournit un flot $d_i$.
- Chaque puits $t_j$ consomme un flot $d_j$.
- Le flot total injecté par les sources est égal au flot total absorbé par les puits :

$$\sum_{i\in S} d_i = \sum_{j\in T} d_j$$

# Problème de Flot à Coût Maximal avec Arc Fictif

## Approche

- Introduction d'un **arc fictif** $(t,s)$ avec :
  - Capacité infinie.
  - Coût $c_{ts} = -1$.
- **Objectif** :

$$\text{Minimiser} \quad z = \sum_{(i,j)\in A} c_{ij} x_{ij} - x_{ts}$$

- Ceci équivaut à **maximiser** $x_{ts}$, le flot total du réseau.

## Formulation Mathématique

$$\begin{aligned} \text{Minimiser} \quad & z = \sum_{(i,j)\in A} c_{ij} x_{ij} - x_{ts} \\ \text{sous contraintes} \quad & \sum_{j\in P_i} x_{ij} - \sum_{j\in B_i} x_{ji} = 0, \quad \forall i \in V \\ & x_{ij} \geqslant l_{ij}, \quad x_{ij} \leqslant u_{ij}, \quad \forall (i,j) \in A^+ \\ & x_{ts} \geqslant 0 \end{aligned}$$

## Interprétation

- L'arc fictif $(t,s)$ permet de transformer le problème en un problème de minimisation avec une fonction objectif linéaire.
- Maximiser le flot sur $x_{ts}$ revient à maximiser le flot total dans le réseau.

# Algorithme de Dijkstra

## Introduction

L'algorithme de Dijkstra est une méthode efficace pour trouver les chemins les plus courts depuis un sommet source $s$ vers tous les autres sommets dans un graphe connexe non orienté et pondéré $G = (V,E)$. Les poids des arêtes représentent les distances ou les coûts, et doivent être non négatifs.

## Notations et Ensembles

$\lambda_{ij}$ : Longueur (ou poids) de l'arête entre les sommets $i$ et $j$, avec $\lambda_{ij} \geqslant 0$.

$N_i$ : Ensemble des voisins (sommets adjacents) du sommet $i$ :

$$N_i = \{j \in V \mid (i,j) \in E\}$$

$EM$ : Ensemble des sommets marqués, c'est-à-dire les sommets pour lesquels le chemin le plus court depuis $s$ a été déterminé.

$EM^c$ : Complémentaire de $EM$, soit l'ensemble des sommets non marqués :

$$EM^c = V \setminus EM$$

$\delta_j$ : Étiquette associée au sommet $j$, représentant la longueur du chemin le plus court de $s$ à $j$ trouvé jusqu'à présent.

## Étapes de l'Algorithme

1. **Initialisation** :
   - ▷ Définir $EM = \{s\}$ et initialiser $\delta_s = 0$.
   - ▷ Pour tout $j \in V \setminus \{s\}$, initialiser $\delta_j = +\infty$.
2. **Étape Itérative** : Tant que $EM^c \neq \emptyset$, répéter les sous-étapes suivantes :

   (a) **Mise à jour des étiquettes** :
   - ▷ Pour chaque sommet $i \in EM$ et chaque voisin $j \in N_i \cap EM^c$, calculer une distance potentielle :

   $$\text{Si } \delta_j > \delta_i + \lambda_{ij}, \text{ alors mettre à jour } \delta_j = \delta_i + \lambda_{ij}$$

   (b) **Sélection du prochain sommet à marquer** :
   - ▷ Trouver le sommet $k \in EM^c$ avec la plus petite étiquette :

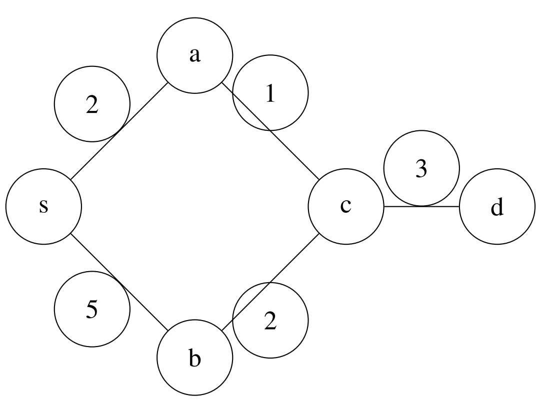
   $$\delta_k = \min_{j \in EM^c} \delta_j$$

   - ▷ Ajouter $k$ à l'ensemble des sommets marqués :
   $$EM = EM \cup \{k\}$$
3. **Terminaison** :
   - ▷ Lorsque tous les sommets sont marqués ($EM = V$), l'algorithme s'arrête. Les étiquettes $\delta_j$ contiennent alors les longueurs minimales des chemins depuis $s$ vers chaque sommet $j$.

## Remarques Importantes

- ▶ L'algorithme de Dijkstra fonctionne correctement uniquement si tous les poids des arêtes $\lambda_{ij}$ sont non négatifs.
- ▶ Il peut être adapté aux graphes orientés en considérant les arcs orientés au lieu des arêtes.
- ▶ L'efficacité de l'algorithme peut être améliorée en utilisant des structures de données appropriées, comme des files de priorité (tas).

## Exemple Illustratif




- ▷ **Initialisation** :
  - $EM = \{s\}$, $\delta_s = 0$, $\delta_a = \delta_b = \delta_c = \delta_d = +\infty$.
- ▷ **Itération 1** :
  - Depuis $s$, mise à jour :

  $$\delta_a = \min(\delta_a, \delta_s + \lambda_{sa}) = 0 + 2 = 2$$

  $$\delta_b = \min(\delta_b, \delta_s + \lambda_{sb}) = 0 + 5 = 5$$

  - Sélection de $a$ (étiquette minimale 2), $EM = \{s, a\}$.
- ▷ **Itération 2** :
  - Depuis $a$, mise à jour :

  $$\delta_c = \min(\delta_c, \delta_a + \lambda_{ac}) = 2 + 1 = 3$$

  - Sélection de $c$ (étiquette minimale 3), $EM = \{s, a, c\}$.
- ▷ **Itération 3** :
  - Depuis $c$, mise à jour :

  $$\delta_d = \min(\delta_d, \delta_c + \lambda_{cd}) = 3 + 3 = 6$$

  - Depuis $b$, aucune mise à jour (étiquette inchangée).
  - Sélection de $b$ (étiquette 5), $EM = \{s, a, c, b\}$.
- ▷ **Itération 4** :
  - Depuis $b$, mise à jour potentielle de $c$, mais $\delta_c$ reste 3 (déjà plus petit).
  - Sélection de $d$ (étiquette 6), $EM = \{s, a, c, b, d\}$.
- ▷ **Terminaison** : Tous les sommets sont marqués. Les étiquettes finales sont :

$$\delta_s = 0, \quad \delta_a = 2, \quad \delta_b = 5, \quad \delta_c = 3, \quad \delta_d = 6$$

## Complexité

- ▶ L'algorithme de Dijkstra a une complexité en $O(|E| + |V| \log |V|)$ lorsqu'il est implémenté avec une file de priorité (tas binaire).