# Introduction

Le problème de transport consiste à envoyer des quantités depuis des fournisseurs vers des clients en respectant :

- chaque fournisseur a une **provision** P[i]
- chaque client a une **demande** C[j]
- chaque trajet (i,j) a un **coût** c[i][j]

On souhaite construire une solution qui répond à toutes les demandes en minimisant le coût total.

Dans notre projet, on fait ça en 2 étapes :

1. On trouve une solution initiale faisable (pas optimale la plupart du temps)
2. Puis on optimise cette solution pour baisser le coût

On compare deux façons de faire :

- **Nord-Ouest -> optimisation**
- **Balas-Hammer -> optimisation**

---

# 1) Nord-Ouest (solution initiale)

**L'objectif :** on remplit le tableau en partant de la case en haut à gauche, sans regarder les coûts.

### Pseudo-code

```
i <- 1 ; j <- 1
Tant que i ≤ n et j ≤ m :
    x <- min(P[i], C[j])
    envoyer x de i vers j
    P[i] <- P[i] - x
    C[j] <- C[j] - x
    Si P[i] == 0 : i <- i + 1
    Sinon si C[j] == 0 : j <- j + 1
```

```
Fin
```

### Complexité

- Très rapide : environ **O(n+m)** (donc ~ **O(n)** si n=m)
- Mais la solution peut être mauvaise, donc l'optimisation derrière peut être longue.

---

# 2) Balas-Hammer (solution initiale)

**L'objectif :** on choisit de manière plus stratégique où envoyer, en utilisant les coûts.

**Principe** :

- on calcule une pénalité par ligne/colonne (différence entre les 2 plus petits coûts)
- on choisit la ligne/colonne avec la plus grosse pénalité
- on alloue sur la case la moins chère correspondante

### Pseudo-code

```
Tant qu'il reste des demandes/provisions :
    Pour chaque ligne i :
        p_ligne[i] <- (2e plus petit coût de la ligne i) - (plus
petit coût)
    Pour chaque colonne j :
        p_col[j] <- (2e plus petit coût de la colonne j) - (plus
petit coût)

    choisir la pénalité max (ligne ou colonne)

    prendre dans cette ligne/colonne la case (i,j) au coût minimal
    x <- min(P[i], C[j])
    envoyer x de i vers j
    mettre à jour P[i], C[j]
    supprimer la ligne ou colonne finie
Fin
```

### Complexité

- Plus lent que Nord-Ouest : souvent proche de **$O(n^3)$** (ça grandit vite)
- Mais donne une meilleure solution initiale → optimisation souvent **plus courte**.

---

## 3) Marche-pied (optimisation)

**Idée :** on améliore une solution faisable en trouvant une case vide qui ferait baisser le coût. On calcule des potentiels u et v, puis des coûts réduits Δ.

- si tous les $\Delta \geq 0$ → solution optimale
- sinon on ajoute la meilleure case (Δ le plus négatif) et on ajuste sur un cycle

### Pseudo-code

```
Répéter :
    calculer les potentiels u[i], v[j] à partir des cases utilisées

    meilleur_delta <- 0
    (p,q) <- rien
    Pour chaque case vide (i,j) :
        delta <- c[i][j] - (u[i] + v[j])
        Si delta < meilleur_delta :
            meilleur_delta <- delta
            (p,q) <- (i,j)

    Si (p,q) n'existe pas :
        Stop (optimal)

    trouver le cycle en ajoutant (p,q)
    theta <- plus petite quantité sur les cases du cycle à diminuer
    ajuster les flux sur le cycle (+theta / -theta)
Jusqu'à optimal
```

### Complexité

- Une itération coûte souvent **$O(n^2)$** (on regarde beaucoup de cases)

- Le nombre d'itérations dépend de la solution initiale :
    - après **Nord-Ouest** → souvent plus d'itérations
    - après **Balas-Hammer** → souvent moins d'itérations

---

## Stratégie d'expérimentation

Pour chaque taille n :

- on génère un problème aléatoire équilibré
- on mesure :
    - θNO : temps Nord-Ouest
    - θBH : temps Balas-Hammer
    - tNO : marche-pied en partant de Nord-Ouest
    - tBH : marche-pied en partant de Balas-Hammer
- on répète 100 fois
- on stocke tout dans un CSV avec aussi :
    - `total_no = θNO + tNO`
    - `total_bh = θBH + tBH`
    - `ratio = total_no / total_bh`

## Conclusion

- **Nord-Ouest** : très rapide, mais donne souvent une solution initiale "pas optimisé" → donc une optimisation beaucoup plus longue.
- **Balas-Hammer** : plus coûteux au début, mais donne une meilleure base → donc une optimisation plus rapide.
- Au final, pour les grandes tailles, Balas-Hammer + marche-pied est souvent plus rentable.