

## TP4: Heuristiques pour le problème de transport

### 1 Objectif du TP

L'objectif de ce TP est de résoudre des problèmes de transport en appliquant les méthodes du **Coin Nord-Ouest** et de **Balas-Hammer**. Dans un premier temps, vous allez calculer manuellement une solution initiale pour chaque méthode. Ensuite, vous décomposerez chaque méthode en plusieurs fonctions intermédiaires afin de simplifier l'implémentation des algorithmes.

### 2 Rappel des Méthodes

#### 2.1 Méthode du Coin Nord-Ouest

La méthode du Coin Nord-Ouest consiste à construire une solution initiale en allouant les offres aux demandes en commençant par la cellule située dans le coin supérieur gauche (nord-ouest) du tableau de transport, puis en avançant soit vers la droite, soit vers le bas, en fonction de l'épuisement de l'offre ou de la demande.

**Pseudocode de la méthode du Coin Nord-Ouest :**

```
Fonction coin_nord_ouest(offre, demande)
    Initialiser i à 0
    Initialiser j à 0
    Tant que i < taille(offre) et j < taille(demande)
        allocation = min(offre[i], demande[j])
        Affecter allocation à la cellule (i, j)
        offre[i] = offre[i] - allocation
        demande[j] = demande[j] - allocation
        Si offre[i] == 0
            i = i + 1
        Sinon si demande[j] == 0
            j = j + 1
        Fin Si
    Fin Tant que
Fin Fonction
```

#### 2.2 Méthode de Balas-Hammer

La méthode de Balas-Hammer consiste à minimiser le coût total de transport en tenant compte des différences associées aux choix de cellules non optimales. À chaque étape, on calcule les différences pour les lignes et les colonnes, puis on choisit la cellule avec la différence maximale pour l'allocation suivante.

**Pseudocode de la méthode de Balas-Hammer :**

```
Fonction balas_hammer(offre, demande, couts)
    Tant que somme(offre) > 0 et somme(demande) > 0
        Pour chaque ligne non satisfaite
            Calculer la différence : différence entre les deux plus petits coûts de la ligne
        Pour chaque colonne non satisfaite
            Calculer la différence : différence entre les deux plus petits coûts de la colonne
        Trouver la ligne ou la colonne avec la différence maximale
        Dans cette ligne ou colonne, choisir la cellule avec le coût minimum
        allocation = min(offre[i], demande[j])
        Affecter allocation à la cellule (i, j)
        offre[i] = offre[i] - allocation
        demande[j] = demande[j] - allocation
        Si offre[i] == 0
            Marquer la ligne i comme satisfaite
        Si demande[j] == 0
            Marquer la colonne j comme satisfaite
    Fin Tant que
Fin Fonction
```

## Exercice 1 : Méthode du Coin Nord-Ouest

Soit le problème de transport donné dans le tableau ci-dessous :

	D1	D2	D3	D4	D5	Offre
O1	7	12	1	5	9	12
O2	15	3	12	6	14	11
O3	8	16	10	12	7	14
O4	18	8	17	11	16	8
Demande	10	11	15	5	4	

### Instructions

#### 1. Calcul de la solution initiale par la méthode du Coin Nord-Ouest :

- À partir du tableau ci-dessus, calculez manuellement la solution initiale en appliquant la méthode du Coin Nord-Ouest. Remplissez un tableau indiquant les quantités allouées à chaque cellule.
- Calculez le coût total associé à cette solution initiale.

#### 2. Conception de fonctions intermédiaires :

- Définissez une fonction `calculer_minimum(offre, demande, i, j)` qui calcule le minimum entre l'offre et la demande à l'étape  $(i, j)$ .
- Créez une fonction `mettre_a_jour(offre, demande, i, j, allocation)` qui met à jour les vecteurs offre et demande après chaque allocation.

#### 3. Implémentation de l'algorithme du Coin Nord-Ouest :

En utilisant les fonctions intermédiaires définies précédemment, compléter la fonction `coin_nord_ouest(offre, demande)` qui implémente l'algorithme complet.

## Exercice 2 : Méthode de Balas-Hammer

Soit le problème de transport donné sous forme matricielle suivante :

	D1	D2	D3	D4	Offre
O1	3	6	4	8	20
O2	3	4	7	9	17
O3	9	4	5	6	13
Demande	12	10	15	13	

### Instructions

#### 1. Calcul de la solution initiale par la méthode de Balas-Hammer :

- Calculez manuellement la solution initiale en appliquant la méthode de Balas-Hammer. Pour chaque itération, calculez les différences pour les lignes et les colonnes, et déterminez la cellule d'allocation optimale.
- Calculez le coût total associé à cette solution initiale.

#### 2. Conception de fonctions intermédiaires :

- Élaborez une fonction `calculer_differences(couts, offre, demande)` qui calcule les différences pour chaque ligne et colonne.
- Définissez une fonction `trouver_position_optimale(couts, differences, offre, demande)` qui détermine la position optimale pour l'allocation actuelle.
- Créez une fonction `mettre_a_jour(offre, demande, i, j, allocation)` pour mettre à jour l'offre et la demande après chaque allocation.

#### 3. Implémentation de l'algorithme de Balas-Hammer :

En utilisant les fonctions intermédiaires définies précédemment, rédigez la fonction `balas_hammer(offre, demande, couts)` qui implémente l'algorithme complet.