Sommaire

Liste des figures :

Introduction générale

La recherche opérationnelle est une approche quantitative permettant de produire de meilleures décisions. Elle fournit des outils pour rationaliser, simuler et optimiser l'architecture et le fonctionnement des systèmes industriels et économiques. Elle propose des modèles pour analyser des situations complexes et permet aux décideurs de faire des choix efficaces et robustes.

Si la recherche opérationnelle, en abrégé RO, est aujourd'hui présente dans la plupart des domaines civils, ses racines sont habituellement attribuées aux services militaires. La seconde guerre mondiale, de part son envergure, créa une besoin urgent d'allouer de manière efficace des ressources limitées aux différentes opérations militaires et aux activités au sein de chaque opération. En particulier, l'organisation militaire britannique, puis américaine, mis à contribution un grand nombre de scientifiques pour gérer ces allocations, et s'occuper d'autres problèmes stratégiques et tactiques.

Parmi les problèmes les plus fréquents, on cite le problème de transport qui est le sujet de notre projet de fin d'étude. Le problème de transport est un modèle important de programmation linéaire qui se pose dans plusieurs contextes et a mérité une attention particulière dans la Recherche Opérationnelle. Il est probablement le problème de programmation linéaire spécial le plus important en termes de fréquence relative avec laquelle il apparaît dans les applications et aussi dans la simplicité de la procédure développée pour sa solution.

Ce document est décomposé en quatre chapitres, et est organisé de la manière suivante : dans le premier chapitre qui est la base de la suite, nous présenterons quelques notions de base de la théorie des graphes et des points généraux de la programmation linéaire.

Dans le second chapitre, nous commencerons par la présentation de problème de transport et sa modélisation en tant qu'un programme linéaire.

Le troisième chapitre s'intéresse à la résolution de problème de transport, nous présenterons les différentes méthodes de détermination d'une solution de base initiale d'un problème de transport, puis les algorithmes d'optimiser cette solution de base initiale.

Le quatrième chapitre est consacré à la description, la programmation en langage C, et la comparaison des méthodes citées au chapitre 3 par l'application à un exemple illustratif.

Nous terminerons ce mémoire en présentant nos conclusions générales et les perspectives de recherche.

Chapitre 1

Notions de base

Introduction

L’objectif de ce chapitre est de rappeler quelques notions de base de la théorie des graphes, et des points généraux de la programmation linéaire, qu’on va utiliser par la suite dans le traitement du problème de transport.

1. Éléments de la Théorie des graphes :

1.1. Un graphe :

a. Définition "intuitive" d'un graphe :

Un *graphe* est un schéma constitué par un ensemble de points et par un ensemble de flèches reliant chacune deux de ceux-ci, dont Les points sont appelés les *sommets* du graphe, et les flèches les *arcs* du graphe.

b. Exemple :

Dans le graphe ci-après, on a :

- un ensemble de sommets :{a,b,c,d,e,f,g,h}.
- un ensemble d'arcs :{1,2,3,4,5,67,8,9}.

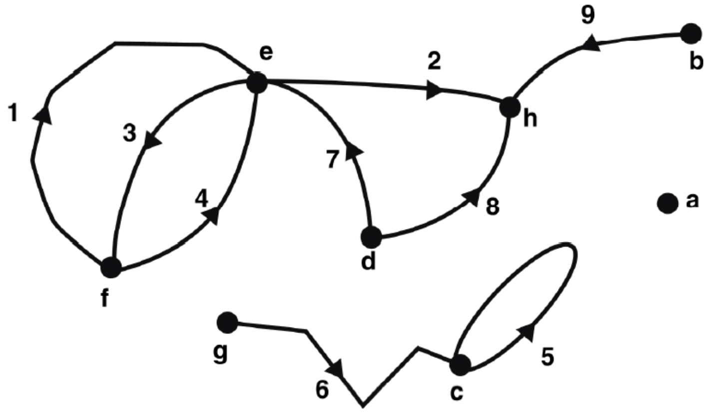


Figure_1: Un exemple d’un graphe.

c. Définition mathématique d'un graphe :

Un graphe G=(X,U) est le couple constitué :

- par un ensemble $X = \{x_1, x_2, \dots x_n\}$, les éléments de X sont appelés les sommets de graphe.
- par une famille $\mathrm{U} = \{\mathrm{u}_1, \mathrm{u}_2, \dots \mathrm{u}_\mathrm{n}\}$ d'éléments du produit cartésien $X \times X = \{(x, y) \mid x \in X, y \in X\}$, les éléments de U sont appelés les arcs de graphe.

1.2. Ordre, orientation et multiplicité d'un graphe :

a. Ordre :

On appelle ordre du graphe le nombre de sommets du graphe.

- Notation :

L'ordre de G est donc le cardinal de X et noté : $|\mathrm{X}|$.

- Exemple :

Le graphe de la figure_1 est d'ordre 8.

b. Orientation :

- Une arête :

Une arête est un ensemble de sommets (ensemble de cardinalité 1 où 2), Une arête est donc un élément de $\mathrm{P}_2(\mathrm{X})$ (l'ensemble des parties à deux éléments de l'ensemble X).

- Graphe non orienté :

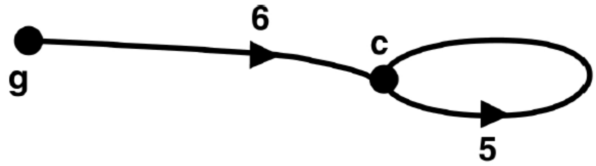
Un "graphe non orienté" sera appelé un multi-graphe et noté $\mathrm{G} = (\mathrm{X}, \mathrm{E})$ avec $\mathrm{E} \subset \mathrm{P}_2(\mathrm{X})$.

c. multiplicité :

- Boucle et extrémités :

Un arc de la forme (x, x) est une boucle. Soit un arc de la forme (x, y), x et y sont appelés les extrémités de l'arc:

- x est l'extrémité initiale de l'arc.
- y est l'extrémité finale de l'arc.



L'arc 5 = (c,c) est une boucle,
g est l'extrémité initiale de 6 et c son extrémité finale

Figure_2: Exemple de boucle et d'extrémités.

➢ p-graphe :

Soit un graphe $G = (X, U)$, si le nombre d'arcs qui va d'un sommet x à un sommet y de X est inférieur où égal à p (pour tous les sommets x et y de X) alors on dit que G est un p-graphe.

$$G = (X, U) \text{ est un p-graphe} \Leftrightarrow \forall (x, y) \in X^2, |\{u \in U / u = (x, y)\}| \leq p$$

➢ Graphe simple :

Un graphe simple est un multi-graphe :
- sans boucles.
- tel qu'il n'y ait jamais plus d'une arête entre deux sommets quelconques.

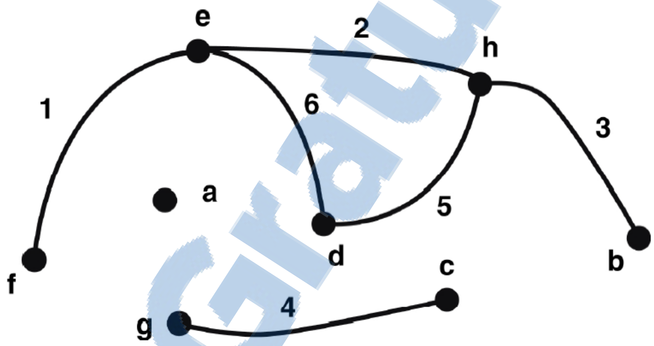


Figure_3: Exemple de graphe simple.

1.3. Qualificatifs des graphes :

a. Sous-graphe partiel :

Soit un graphe G=(X,U) quelconque. Soit A ⊂ X, alors le sous graphe engendré par A est le graphe G_A dont les sommets sont les éléments de A et dont les arcs sont les arcs de G ayant leurs deux extrémités dans A.

Soit V⊂ U alors le graphe partiel engendré par V est le graphe (X,V) ayant le même ensemble X de sommets que G, et dont les arcs sont les arcs de V(on élimine de G les arcs de U\V).

Un sous-graphe partiel de G est un sous-graphe d'un graphe partiel de G où un graphe partiel d'un sous-graphe de G.

b. Graphe biparti complet :

➢ Un graphe complet:

Un graphe est complet si pour toute paire de sommets x et y, il existe au moins un arc de la forme (x,y) où de la forme (y,x).

- Un graphe biparti :

Un graphe est biparti si l'ensemble de ses sommets peut être partitionné en deux classes X_1et X_2 de sorte que deux sommets de la même classe ne soient jamais voisins.

Soit un graphe biparti, si pour tout $x_1 \in X_1$, et pour tout $x_2 \in X_2$, on a $m_G(x_1, x_2) \geq 1$, alors le graphe est biparti complet.

- Notation :

- Un graphe biparti peut être noté G=(X_1, X_2, U).
- Un graphe simple biparti-complet avec $|x_1|$=p et $|x_2|$=q est noté $K_{p,q}$.

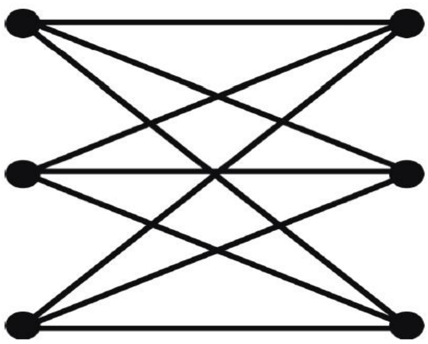


Figure_4: Graphe biparti complet $K_{3,3}$.

1.4. Matrices associées à un graphe :

a. Matrice d'incidence sommet-arc :

- Définition :

La matrice d'incidence sommets-arcs d'un graphe G=(X,U) sans boucle est une matrice telle que chaque colonne correspond à un arc de G et chaque ligne à un sommet de G; si u=(i,j) ∈ U, la colonne u a tous ses termes nuls, sauf :

$$\begin{cases} a_{ij} = +1 \\ a_{ji} = -1 \end{cases}$$

Si on considère une ligne correspondant au sommet alors on retrouve aisément l'ensemble des arcs incidents au sommet :

$$\begin{cases} w^+(i) = \{u = (i, j) \text{ tq}: a_{ij} = +1\} \\ w^-(i) = \{u = (i, j) \text{ tq}: a_{ij} = -1\} \end{cases}$$

- <u>Exemple :</u>

On donne comme exemple la matrice d'incidence sommets-arcs du graphe partiel sans boucle engendré par l'ensemble des arcs du graphe de la figure_1 privé de l'arc 5 (la boucle) :

$$
\begin{array}{c|c}
 & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 6 & 7 & 8 & 9 \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{array} &
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & +1 & +1 & 0 \\
-1 & +1 & +1 & -1 & 0 & -1 & 0 & 0 \\
+1 & 0 & -1 & +1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & -1 & -1
\end{pmatrix}
\end{array}
$$

b. Matrice d'adjacence où d'incidence sommets-sommets :

- <u>Définition :</u>

La matrice d'adjacence où d'incidence sommets-sommets d'un graphe G=(X,U) est une matrice carrée où chaque ligne correspond à un sommet de G et chaque colonne correspond également à un sommet de G. Le terme général de la matrice est : $a_{ij} = m_G^+(x_i, x_j)$.

- <u>Note:</u>

Contrairement à la matrice d'incidence sommets-arcs les boucles peuvent être représentées grâce à cette matrice.

- <u>Exemple</u>

On donne comme exemple la matrice d'incidence sommets associée au graphe de la figure_1.

$$
\begin{array}{c}
\\
a\\ b\\ c\\ d\\ e\\ f\\ g\\ h
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array}\\
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1\\
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

1.5. Vocabulaire lié à la connexité :

a. Chaîne, chemin, longueur :

- Une chaîne de longueur q > 0 est une séquence $\mu=(u_1, u_2, \ldots, u_q)$ d'arcs de G telle que chaque arc de la séquence est une extrémité en commun avec l'arc précédent, et l'autre extrémité en commun avec l'arc suivant. Le nombre d'arcs de la séquence est la longueur de la chaîne.

- Une chaîne élémentaire est une chaîne ne rencontrant pas deux fois le même sommet.

- Une chaîne simple est une chaîne n'utilisant pas deux fois le même arc.

- Un chemin de longueur q < 0 est une chaîne telle que pour tout i < q l'extrémité finale de u_i coïncide avec l'extrémité initiale de u_{i+1} (en d'autres termes, u_{i+1} est un successeur de u_i).

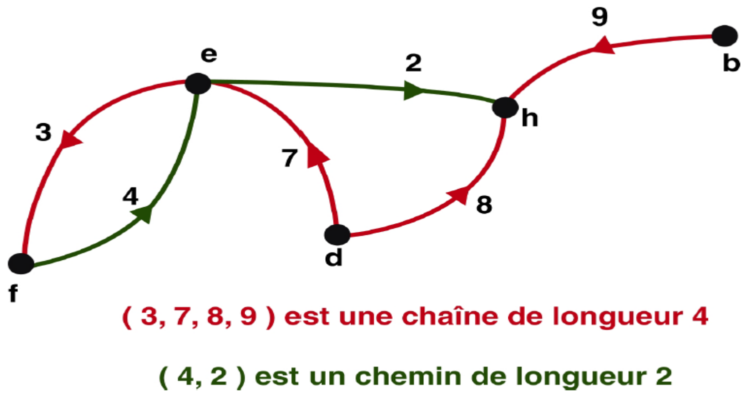


Figure_5: Exemple de chaine et de chemin.

b. Connexité :

Un graphe connexe est un graphe tel que pour toute paire de sommets x et y , il existe une chaîne reliant x et y .

La relation "x=y où x≠y et il existe une chaîne reliant x et y " est une relation d'équivalence (notée $x \equiv y$) car :

- $x \equiv x$ (réflexivité)
- $x \equiv y \Longrightarrow y \equiv x$ (symétrie)
- $x \equiv y$ et $y \equiv z \Longrightarrow x \equiv z$ (transitivité)

Les classes d'équivalences de cette relation sont appelées les composantes connexes du graphe. Le nombre de connexité du graphe est simplement le nombre de composantes connexes.

2. Programmation linéaire :

La programmation linéaire est un outil très puissant de la recherche opérationnelle. C'est un outil générique qui peut résoudre un grand nombre de problèmes. En effet, une fois un problème modélise sous la forme d'équations linéaires, des méthodes assurent la résolution du problème de manière exacte.

On distingue dans la programmation linéaire, la programmation linéaire en nombres réels, pour laquelle les variables des équations sont dans IR+ et la programmation en nombres entiers, pour laquelle les variables sont dans IN. Bien entendu, il est possible d'avoir les deux en même temps. Cependant, la résolution d'un problème avec des variables entières est nettement plus compliquée qu'un problème en nombres réels.

Une des méthodes les plus connues pour résoudre des programmes linéaires en nombre réels est la méthode du Simplex. En théorie, elle a une complexité non polynomiale et est donc supposée peu efficace. Cependant, en pratique, il s'avère au contraire qu'il s'agit d'une bonne méthode.

Un programme linéaire est la maximisation où la minimisation d'une fonction linéaire sous des contraintes linéaires.

2.1. Forme générale d'un programme linéaire :

$$\begin{cases} \text{max où min} \sum_{j=1}^{n} c_j x_j & (1) \\ \forall i = 1, \dots, m \ : \sum_{j=1}^{n} a_{ij} x_j \ \ \le, = \text{où} \ \ge b_i & (2) \\ \forall j = 1, \dots, n \ \ xj \ge 0 & (3) \end{cases}$$

(1) : fonction objective.

(2) : m contraintes linéaires.

(3) : contraintes de positivité.

2.2. Formes matricielles classiques et conventions :

Notons par $x = (x_1, x_2, \dots, x_n)^T$ le vecteur des variables. $b = (b_1, b_2, \dots, b_m)^T$ le second membre des contraintes, $c = (c_1, c_2, \dots, c_n)^T$ le vecteur coût où profit associé aux variables et A la matrice $m \times n$ des a_{ij} .

$$\begin{cases} \text{forme canonique:} \\ \max z = cx \\ Ax \leq b \\ x \geq 0. \end{cases} \qquad \begin{cases} \text{forme standard:} \\ \max z = cx \\ Ax = b \\ x \geq 0. \end{cases}$$

La forme canonique avec des contraintes ≤ s'utilise dans la représentation graphique, et la forme standard avec des contraintes égalité s'utilise dans la résolution algébrique.

2.3. Interprétation économique :

Un programme linéaire a une interprétation économique très large :

- Un acteur économique qui exerce n activités avec des intensités x_j à déterminer.
- Ces activités utilisent m ressources.
- La quantité a_{ij} de ressources i nécessaires pour exercer l'activité j avec une intensité 1.
- On connait le profit (en maximisation) et le coût (en minimisation).

c_j Correspond a une intensité 1 de l'activité j.

2.4. La méthode de simplexe:

L'algorithme du simplexe a été introduit par George Dantzig à partir de 1946. C'est un algorithme de résolution des problèmes d'optimisation linéaire.

Il consiste à minimiser une fonction linéaire de n variables réelles,

$$x = (x_1, x_2, \dots, x_n) \in IR \mapsto c^T x := \sum_{i=1}^{n} c_i x_i$$

Où $c = (c_1, \dots, c_n) \in IR$ sur un ensemble défini au moyen de contraintes affines (ou linéaires) d'égalité et d'inégalité. L'ensemble admissible du problème est donc un polyèdre convexe.

➢ <u>Les deux phases de la méthode du simplexe :</u>

- **Phase 1 (Initialisation) :** Trouver une solution de base réalisable (ou bien détecter l'impossibilité).
- **Phase 2 (Progression) :** On passe d'un sommet à un sommet voisin pour augmenter la fonction objectif.

PL sous forme standard :

$$\max_{x \in IR^n} [\, F(x) = c^T x \,]$$

$$\begin{cases} Ax = b \\ x \geq 0 \end{cases}$$

On dispose d'une base B et d'une solution de base réalisable $\bar{x}$ avec (avec une permutation prés des colonnes de A) :

$$A = (A_B | A_H) \qquad \text{et} \qquad \bar{x} = \begin{pmatrix} \bar{x}_B \\ \bar{x}_H \end{pmatrix}$$

Où A_B matrice $m \times m$, inversible (variables de base).

A_H matrice $m \times (n - m)$ (variables hors − base).

- L'objectif :

On veut trouver une autre base B^* et une solution de realisable $\bar{x}^*$ telles que $\bar{x}^*$ est meilleur que $\bar{x}$ c-a-d : $F(\bar{x}^*) > F(\bar{x})$

- Principe de la méthode du simplexe :

Faire rentrer une variable hors-base dans la nouvelle base (variable entrante) et faire sortir à la place une variable de base (variable sortante).

- Variable entrante - calcul des coûts réduits :

Fonction objectif F exprimée en fonction des variables hors-base.
Ensemble des solutions réalisables $D_R = \{x \in IR^n | Ax = b,\ x \geq 0\}$

Proposition :

Pour tout $x \in D_R$ on a : $F(x) = F(\bar{x}) + L_H^T x_H$

Où $L_H^T = c_H^T - c_B^T A_B^{-1} A_H$ est le vecteur des coûts réduits.

Variable entrante

- Si les coûts réduits sont tous négatifs i.e. $L_H \leq 0$, il n'est alors pas possible d'augmenter la fonction objectif F: l'algorithme se termine normalement c'est-adire qu'on a trouvé une solution de base réalisable $\bar{x}$ optimale.
- Dans le cas contraire (i.e $\exists (L_H)_i > 0$) on a intérêt a faire entrer dans la base, la variable hors-base qui a le coût réduit positif le plus grand possible.

On note $e \notin B$ l'indice de la variable entrante.

On choisit e tel que : $(L_H)_e = \max_j\{(L_H)_j, (L_H)_j > 0\}$

Ce qu'on note par : $e = \arg\max_j\{(L_H)_j, (L_H)_j > 0\}$

<u>Variable sortante :</u>

Une fois l'indice e choisi, il faut déterminer la variable qui doit quitter la base. En maintenant la relation $Ax = b$ avec $x \geq 0$, on augmente la variable entrante x_e jusqu'a annuler une des variables de base. Cette variable sera alors la variable sortante.

$Ax = b \Leftrightarrow A_B x_B + A^e x_e = b$ Où A^e désigne la e-ème colonne de A

$\Leftrightarrow x_B = A_B^{-1}(b - A^e x_e)$

$\Leftrightarrow x_B = \bar{x}_B - A_B^{-1} A^e x_e$

$\Leftrightarrow x_B = \bar{x}_B - z x_e$

Avec : $z = A_B^{-1} A^e \in IR^m$

On doit avoir **:** $x_B = \bar{x}_B - z x_e \geq 0$

- Si $z \leq 0$, on peut augmenter x_e autant qu'on veut, on aura toujours la positivité de la variable de base x_B. La fonction objective n'est pas majorée sur $D_R(\max F = +\infty) \Longrightarrow$ arrêt de l'algorithme.

- Sinon (i.e. il existe $z_i > 0$), pour avoir la positivitee $(\bar{x}_B)_I - z_i x_e \geq 0$ pour tout i, on choisit la variable sortante x_s pour laquelle le rapport $(\bar{x}_B)_i / z_i$ pour $i = 1, \dots, m$ avec $z_i > 0$, est le plus petit possible :

Variable sortante (indice) :

$$s = arg \min_i\{\frac{(\bar{x}_B)_i}{z_i}, z_i > 0\}$$

On a, dans ce cas, $x_s = 0 \; et \; x_B \geq 0$.

<u>Remarque :</u> La valeur de la variable entrante est donnée par :

$$x_e = \min_i\{\frac{(\bar{x}_B)_i}{z_i}, z_i > 0\}$$

- <u>Méthode du simplexe en phase 2 (progression) :</u>

1. Calcul des variables de base réalisables :
   Étant donné $A = (A_B|A_H)$, on calcule $\overline{x}_B = A_B^{-1}b \geq 0$.

   Calcul des coûts réduits :
   $$L_H^T = c_H^T - c_B^T A_B^{-1} A_H \qquad (F(x) = F(\bar{x}) + L_H^T x_H)$$
   Si $L_H \leq 0$ alors $\overline{x}_B$ est une solution optimale (→arrêt de l'algorithme.).

2. Variable entrante : e=$\text{argmax}_j\{(L_H)_j, (L_H)_j > 0\}$

3. Variable sortante :
   - Calcul de $z = A_B^{-1}A^e$ puis
   - $s = arg\ \min_i \left\{\frac{(\overline{x}_B)_i}{z_i}, z_i > 0\right\}$.

4. On obtient une nouvelle base $\widetilde{B}$ et une nouvelle matrice $A_{\widetilde{B}}$ dans laquelle la colonne A^e on remplace la colonne A^e, on calcule de $A_{\widetilde{B}}^{-1}$ et retour en 1.

Chapitre 2

Problème de transport

Introduction :

Ce chapitre a l'objectif de présenter et de modéliser le problème de transport par ses différents formulations, Il s'agit d'un type de problème de programmation linéaire qui peut être énoncer comme suit: " Comment transporter aux moindres coûts entre m origines X_i={$x_1, \dots, x_m$} et n destinations Y_j={$y_1, \dots, x_n$}. Les disponibilités a_i $(i = 1, \dots, m)$ existant aux origines x_i $(i = 1, \dots, m)$ afin de satisfaire les demandes $b_j(j = 1, \dots, n)$ des destinations y_j $(j = 1, \dots, n)$, étant données $m \times n$ coûts de transport c_{ij} . "

1. Positionnement de problème :

Dans un problème de transport, nous avons certaines origines, ce qui peut représenter les usines où nous avons produit des articles et fourni une quantité requise de produits à un certain nombre de destinations. Cela doit être fait de manière à maximiser le profit où à minimiser le coût, Ainsi, nous avons les lieux de production comme origines et les lieux d'approvisionnement en tant que destinations.

- Caractéristiques:

- Un problème de transport est un problème de minimisation du coût de transport.
- C'est un problème qui s'inscrit dans la lignée de programmes linéaires.
- Le problème de transport a pour but d'acheminer aux moindres coûts des marchandises depuis m origines vers n destinations.

2. Modèle mathématique :

Supposons qu'une entreprise ait m entrepôts et n points de vente, Un seul produit doit être expédié des entrepôts aux points de vente. Chaque entrepôt (*origine*) a un niveau d'approvisionnement donné (*disponibilité*), et chaque Point de vente (*destination*) a un niveau de demande (*demande*) donné. On nous donne également le coût de transport entre chaque paire d'entrepôt et de destination, et ces coûts sont supposés linéaires.

Plus explicitement, Les hypothèses sont les suivantes:

- La disponibilité de chaque entrepôt i est : a_i, où i = 1,2,3 ... m.
- La demande de chaque destination j est : b_j, où j = 1,2,3 ... n.
- Le coût de transporter une unité du produit de l'entrepôt i à la destination j est égal à c_{ij} .

Où i=1,2,3 ... m et j=1,2,3, ... n. Le coût total d'une expédition est linéaire en taille d'expédition.

2.1. Les variables de décision :

Les variables du modèle de programmation linéaire (LP) du problème de transport sont des entiers naturels représentant des unités transportées d'une source vers une destination.
Les variables de décision sont les suivantes:

x_{ij} : La quantité à transporter de source i à la destination j.
Où i = 1,2,3 ... m et j =1,2,3, ... n. Il s'agit d'un ensemble de $m \times n$ variables.

2.2. La fonction objective :

La fonction objective contient des coûts associés à chacune des variables. C'est une minimisation de problème.

Considérons le transport de source i à la destination j, $\forall\, i \in \{1, \dots, m\}, \forall\, j \in \{1, \dots, n\}$:
c_{ij} : Le coût de transport par unité.
x_{ij} : La quantité à transporter.

Puisque nous supposons que la fonction coût total est linéaire, Le coût total de cette expédition est donné par $c_{ij} \times x_{ij}$ En sommant sur tout i et j, on obtient le coût global de transport pour tous les entrepôts.

C'est-à-dire que notre fonction objective est:

$$\boldsymbol{min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}} \qquad (1.1)$$

2.3. Les contraintes :

Les contraintes sont les conditions qui obligent à satisfaire la demande et épuiser la disponibilité.
Dans un Problème de transport, il existe une contrainte pour chaque nœud.
Posons : a_i désigne une capacité d'une source (*disponibilité*) et b_j désigne le besoin d'une destination (*demande*).

Les contraintes sont :

1) La disponibilité à chaque source doit être épuisée : $\sum_j^n x_{ij} = a_i$, $i = 1, \dots, m$ (1.2)
2) La demande à chaque destination doit être satisfaite: $\sum_i^m x_{ij} = b_j$, $j = 1, \dots, n$ (1.3)
3) Non négativité des quantités : $x_{ij} \geq 0 \quad \forall i = 1, \dots, m \ ; \ \forall j = 1, \dots, n$ (1.4)

2.4. Formulation mathématique :

Le modèle de transport deviendra alors:

$$\begin{cases} \text{minimiser } z = \sum_i^m \sum_j^n c_{ij} x_{ij} \\ \sum_j^n x_{ij} = a_i \ \ \forall i \in \{1, \dots, m\} \\ \sum_i^m x_{ij} = b_j \ \ \forall \in \{1, \dots, n\} \\ a_i \in IN \ \ \forall i \in \{1, \dots, m\} \\ b_j \in IN \ \ \forall j \in \{1, \dots, n\} \\ x_{ij} \in IN \ \ \forall (i,j) \in \{1, \dots, m\} \times \{1, \dots, n\} \end{cases} \tag{1.5}$$

Il s'agit d'un programme linéaire avec $m \times$ n variables de décision, $m + n$ contraintes fonctionnelles et $m \times n$ contraintes non négatives.

m : Nombre de sources.
n : Nombre de destinations.
a_i : Disponibilité de la i-ème source (en tonnes, livres, litres, etc.)
b_j : Demande de la j-ème destination (en tonnes, livres, litres, etc.)
c_{ij} : Coût unitaire de transport entre i-ème source et j-ème destination (en $, kilomètres, miles, etc.)
x_{ij} : Quantité transportée entre i-ème source et j-ème destination (en Tonnes, livres, litres etc.)

- Condition nécessaire et suffisante :

Une condition nécessaire et suffisante pour l'existence d'une solution réalisable au problème transport est que :

$$\sum_i^m a_i = \sum_j^n b_j \ \ \forall (i,j) \in \{1, \dots, m\} \times \{1, \dots, n\} \tag{1.6}$$

- Remarque :

L'ensemble des contraintes $\sum_j^n x_{ij} = a_i \ \forall i \in \{1, \dots, m\}$ et $\sum_i^m x_{ij} = b_j \ \forall j \in \{1, \dots, n\}$ représente m+n équations dans m×n variables non négatives. Chaque variable x_{ij} apparaît exactement en deux contraintes, l'une est associée à l'origine et l'autre est associée à la destination.

3. Problème de transport non équilibré :

Si

$$\sum_i^m a_i \neq \sum_j^n b_j \tag{1.7}$$

Ce problème du transport est connu comme un problème de transport déséquilibré. On a deux Cas :

$\sum_i^m a_i > \sum_j^n b_j$ Où $\sum_i^m a_i < \sum_j^n b_j$

On pourra se ramener à l'énoncé précédent de la manière suivante :

- Si $\sum_i^m a_i > \sum_j^n b_j$ il suffit d'introduire une destination fictive y_{n+1} de coût de transport égale à zéro entre x_i et y_{n+1} (i=1,⋯,m) dont la demande :

$$b_{n+1} = \sum_i^m a_i - \sum_j^n b_j \tag{1.8}$$

- Si $\sum_i^m a_i < \sum_j^n b_j$ il suffit d'introduire une source fictive x_{n+1} de coût de transport égale à zéro entre y_j et x_{m+1} (j=1,⋯,n) dont la disponibilité :

$$a_{m+1} = \sum_j^n b_j - \sum_i^m a_i \tag{1.9}$$

4. Tableau de transport :

Le problème de transport peut être décrit en utilisant un modèle mathématique de programmation linéaire, et habituellement, il apparaît dans un tableau de transport.
Le modèle d'un problème de transport peut être représenté sous forme de tableau concis avec tous les paramètres pertinents.

Le tableau de transport (Un problème de transport typique est représenté sous forme de matrice standard), où la disponibilité d'approvisionnement (a_i) à chaque source est affichée dans la colonne droite du tableau, et les demandes de destination (b_j) sont affichées dans la ligne inférieure.

Chaque cellule représente une voie, Le coût de transport unitaire (c_{ij}) est indiqué dans le coin supérieur droit de la cellule, la quantité de matériel transporté est affichée au centre de la cellule, Le tableau de transport exprime implicitement les contraintes de l'offre et de la demande et le coût de transport entre chaque source et destination.

Destination : → Source : ↓	D_1	D_2	... D_j ...	D_n	Disponibilités
S_1	C_{11} x_{11}	C_{12} x_{12}		C_{1n} x_{1n}	a_1
S_2	C_{21} x_{21}	C_{22} x_{22}		C_{2n} x_{2n}	a_2
... S_i ...			C_{ij} x_{ij}		a_i
S_m	C_{m1} x_{m1}	C_{m2} x_{m2}		C_{mn} x_{mn}	a_m
Demandes	b_1	b_2	... b_j ...	b_m	$\sum a_i$ $\sum b_j$

Figure_11: tableau de transport

5. Réseau de transport :

Graphiquement, le problème du transport est souvent visualisé comme un réseau avec *m* nœuds sources, *n* nœuds destinations et un ensemble de m×n "arcs orientés " Ceci est représenté dans la figure_12.

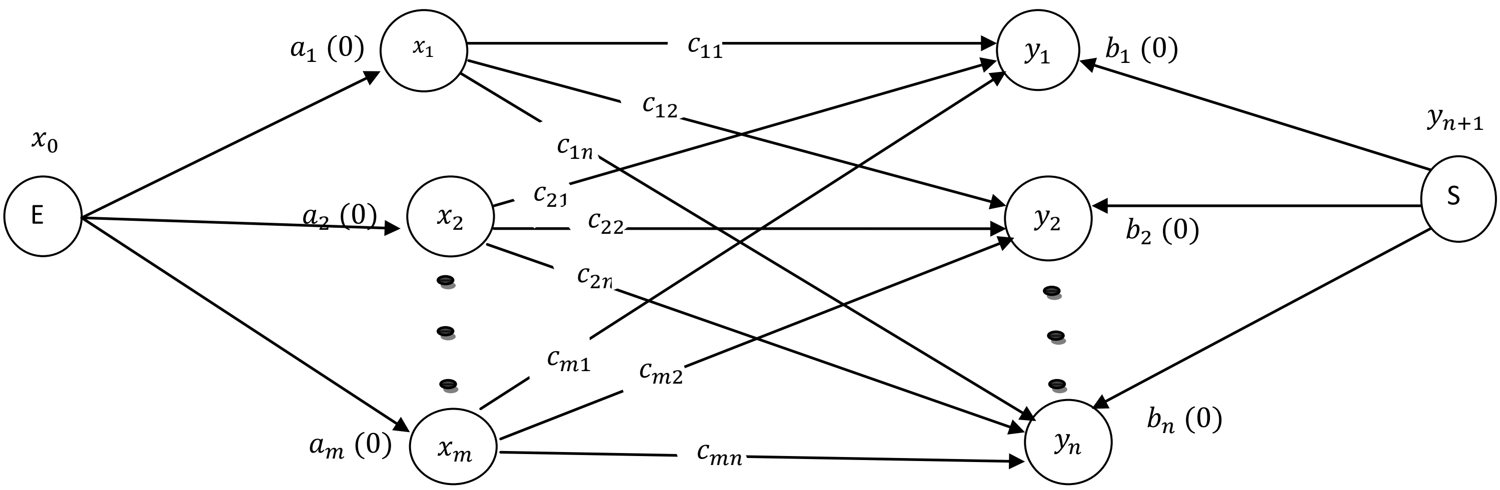


Figure_12: Réseau de transport associé

Dans la figure_12, il y a $x_1 \dots x_m$ sources et $y_1 \dots y_n$ destinations. Les arcs orientés montrent des flux de transport de source vers destination. Chaque destination est liée à chaque source par un arc, Le nombre ($c_{11} \dots c_{mn}$) au-dessus de chaque arc représente le coût du transport sur cette route.

Les problèmes avec la structure ci-dessus se posent dans de nombreuses applications. Par exemple, les sources pourraient représentent des entrepôts, des puits,$\cdots$etc. et les destinations pourraient représentent des populations, des clients,...etc.

6. Dégénérescence en problème de transport :

La dégénérescence existe dans un problème de transport lorsque le nombre de cellules remplies est inférieur au nombre de lignes plus le nombre de colonnes moins un (m + n – 1). La dégénérescence peut être observée soit lors de l'attribution initiale lorsque la première entrée dans une ligne où une colonne satisfait à la fois aux exigences de la ligne et de la colonne où lors de l'application d'une méthode de résolution de problème de transport, lorsque les valeurs ajoutées et soustraites sont égales.

Le transport avec m-origines et n-destinations peut avoir (m + n – 1) variables de base positives, sinon la solution de base dégénèrera, Donc chaque fois que le nombre de cellules basiques est inférieur à m + n - 1, le problème du transport est dégénéré.
Pour résoudre la dégénérescence, les variables positives sont augmentées par autant de variables à valeur nulle que nécessaire pour compléter les (m + n – 1) variables de base.

Chapitre 3

Résolution de problème de transport

Introduction :

Le problème de transport est ainsi un programme linéaire et peut donc être résolu par des méthodes du simplexe. Cependant elles existent des méthodes qui ont même principe de résolution de simplexe mais plus adaptées, Comme toute application de la méthode du simplexe à la résolution de ce type de problème, nécessite une solution de base initiale. La détermination de cette solution et son amélioration est l'objectif de ce chapitre.

1. Structure de la résolution de problème de transport :

Considérons un problème de transport impliquant m origines et n destinations. Étant donné que la somme des disponibilités d'origine est égale à la somme des demandes de destination, une solution réalisable existe toujours. La (m+n)-nième contrainte est redondante et peut donc être supprimée. Cela signifie également qu'une solution de base réalisable pour un problème de transport peut avoir au plus (m+n–1) composants strictement positifs, Sinon la solution dégénérera.

Il est toujours possible d'assigner une solution réalisable initiale à un problème de transport. De telle sorte que les exigences des destinations soient satisfaites. Cela peut être réalisé soit par une inspection, soit par des règles simples. Nous commençons par imaginer que la table de transport est vide, c'est-à-dire initialement tout $\boldsymbol{x_{ij} = 0}$. Les procédures les plus simples pour l'allocation initiale seront discutées dans la section suivante.

1.1. Solution de base réalisable :

- <u>Définition :</u>

On appelle solution de base réalisable une solution vérifiant les contraintes du problème comportant exactement (m-1)×(n-1) flux nuls.

Si le nombre d'allocations dans les solutions de bases réalisables est inférieur à (m + n – 1), on appelle une solution de base dégénérée.

- <u>Interprétation en termes de théories de graphes de la notion de solution de base :</u>

Imaginons que nous disposons d'un algorithme qui permet à toute étape de satisfaire une demande ou épuiser une disponibilité.

À chaque étape de l'algorithme $\varphi_{ij} = \min(a', b')$ $(a'_i, b'_j \neq 0, \varphi_{ij} \in \mathrm{IN})$ avec :
a'_i= disponibilité restante dans x_i.
b'_j= demande insatisfaisante dans y_j.

Si $a'_i \neq b'_j$ (sauf la dernière étape où on a $a'_i = b'_j$) nous aurons choisi (m+n – 1) flux (φ_{ij}) non nuls et nous obtenons une solution de base :
$(m \times n - (m + n - 1) = (m - 1)(n - 1)$ flux nuls).

1.2. Solution optimale :

- <u>Définition :</u>

Une solution réalisable (pas nécessairement de base) est considérée optimale si elle minimise le coût total du transport.

- <u>Vocabulaire :</u>

- Circuit: un circuit est une séquence de cellules (dans le tableau de transport équilibré) tel que :
    1) Il commence et se termine avec la même cellule.
    2) Chaque cellule de la séquence peut être connectée au membre suivant par une ligne horizontale ou verticale dans le tableau.

- Allocation: le nombre d'unités d'articles transportés d'une source à une destination enregistrée dans une cellule dans le tableau de transport.

- Variables de base: les variables dans une solution de base dont les valeurs sont obtenues en tant que solution simultanée du système d'équations qui constituent les contraintes fonctionnelles.

1.3. Organigramme de résolution pour le problème de transport :

On résume la résolution de problème de transport sous forme d'organigramme suivant :

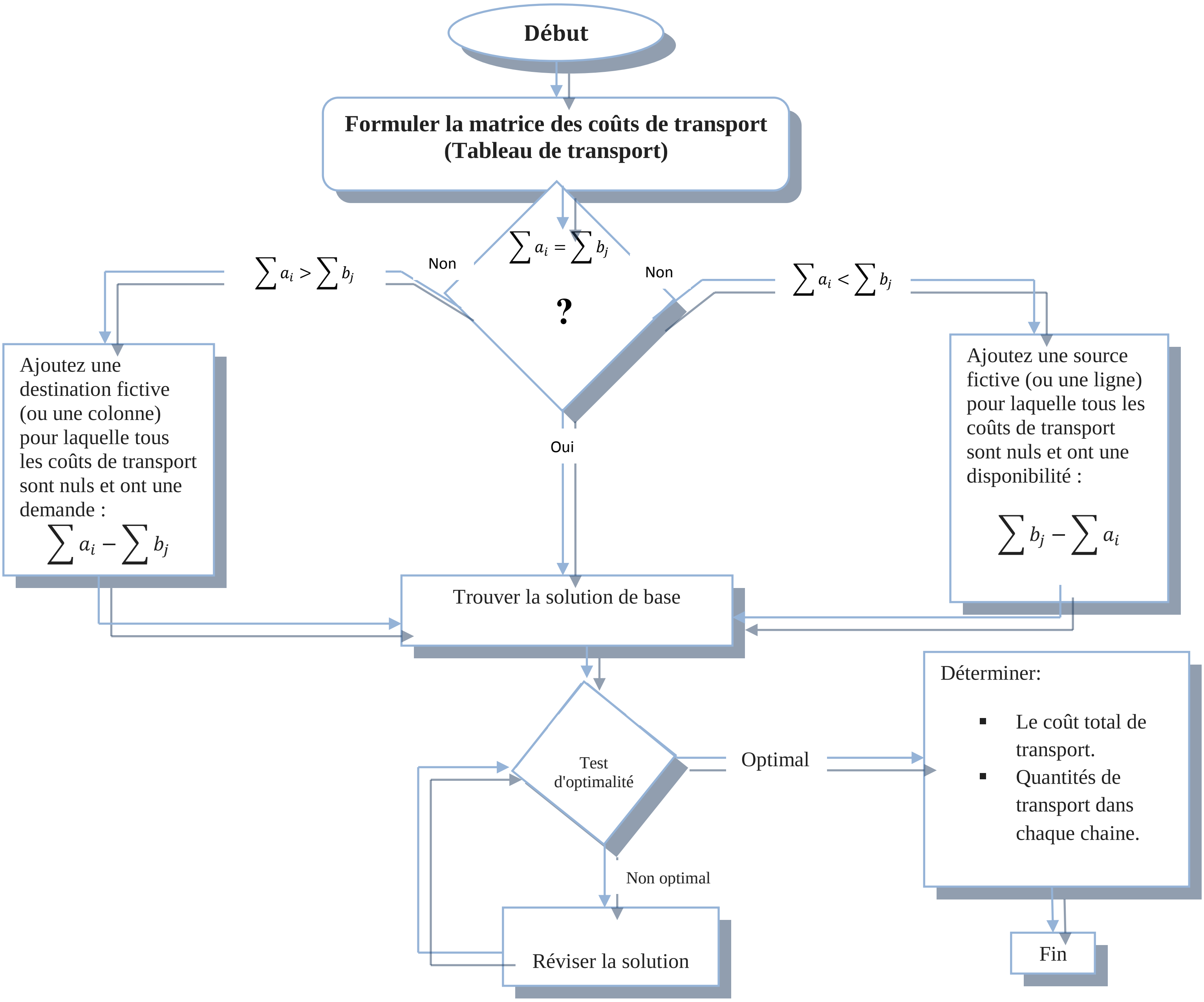


Figure_13: Organigramme de la résolution de problème de transport.

➢ Description sommaire d’organigramme de résolution :

1) Tout d'abord, le problème est formulé comme une matrice de transport.
2) Vérifiez si un modèle de transport est équilibré?
3) Sinon, ajoutez un mannequin à l'offre ou à la demande pour équilibrer le modèle de transport.
4) Trouvez la solution de base initiale du problème de transport.
5) Vérifiez si la solution est optimale ? Si la solution n'est pas optimale, passez à 4.
6) Lorsque la solution optimale est obtenue, nous calculons le coût total du transport et nous avons également transporté la quantité respective demandée à son destinataire.

1.4. Algorithme général de résolution de problème de transport :

Les modèles de transport ne commencent pas à l'origine où toutes les valeurs de décision sont nulles, Ils doivent plutôt recevoir une solution de base réalisable initiale.

L'algorithme de résolution à un problème de transport peut se résumer en étapes suivantes :

Étape 1 : Formuler et configurer le problème sous la forme matricielle :

La formulation du problème de transport est similaire à la formulation du problème PL. Ici, la fonction objective est le coût total du transport et les contraintes sont l'offre et la demande disponibles à chaque source et destination, respectivement.

Étape 2 : Obtenir une première solution de base réalisable :

Cette solution de base initiale peut être obtenue en utilisant l'une des méthodes suivantes:

- Méthode de Coin Nord-Ouest.
- Méthode du Coût Minimum.
- Méthode d'Approximation de Vogel.

La solution obtenue par l'une des méthodes ci-dessus doit satisfaire les conditions suivantes :

1. La solution doit être réalisable, c'est-à-dire qu'elle doit satisfaire toutes les contraintes de l'offre et de la demande.
2. Le nombre d'attribution positive (les cases allouées) doit être égal à m + n - 1, où m est le nombre de lignes et n est le nombre de colonnes.

La solution qui satisfait les conditions mentionnées ci-dessus est appelée une solution de base non dégénérée.

Étape 3 : Tester la solution de base initiale pour l'optimalité :

L'utilisation de l'une des méthodes suivantes pour tester l'optimalité de la solution de base initiale obtenue:

- Méthode Stepping Stone.
- Méthode de distribution modifiée.

Si la solution est optimale arrêtez, sinon déterminez une nouvelle solution améliorée.

Étape 4. Mise à jour de la solution

Répétez l'étape 3 jusqu'à atteindre la solution optimale.

2. Méthodes de détermination de solution de base initiale :

2.1. Méthode de Coin Nord-Ouest :

- Définition :

Il s'agit de la méthode heuristique la plus facile à appliquer pour calculer une solution de base réalisable d'un problème de transport où les variables de base sont sélectionnées à partir du coin nord-ouest (c-à-d Coin supérieur gauche du tableau de transport).

- Principe :

La méthode commence à la case du coin du nord-ouest (itinéraire). L'avantage principal de la méthode du coin nord-ouest est qu'il est très simple et facile à appliquer. Son inconvénient majeur, cependant, est qu'il n'est pas sensible aux coûts et par conséquent produit de mauvaises solutions initiales, elle donne souvent des solutions éloignées de l'optimum.

- Les étapes de la méthode de Coin Nord-Ouest:

1) Sélectionner la case dans le coin supérieur à gauche, sous réserve des conditions d'offre et de demande.
2) Allouer autant que possible à la prochaine case possible adjacente.
3) Répétez l'étape 2 jusqu'à ce que toutes les exigences du tableau soient respectées.

- Algorithme de coin nord-ouest:

1) i=1, j=1.
2) $\varphi_{ij} = \min(a_i, b_j)$ si $\varphi_{ij} = a_i$ passer à (3) sinon passer à (4).
3) Poser $b_j = b_j - a_i$ et i=i+1, si $i \leq n$ passer à (2) sinon FIN.
4) Poser $a_i = a_i - b_j$ et j=j+1, si $j \leq n$ passer à (2) sinon FIN.

2.2. Méthode du Coût Minimum :

- Définition :

La méthode du Coût Minimum est une méthode pour calculer une solution de base réalisable d'un problème de transport où les variables de base sont choisies en fonction du coût unitaire du transport.
La méthode du coût minimum trouve une meilleure solution de départ en se concentrant sur les coûts de transport les moins chers.

➢ <u>Principe :</u>

La méthode commence par affecter autant que possible à la case avec le coût unitaire de transport le plus petit. Ensuite, la ligne où la colonne satisfaite est dépassée et les montants de l'offre et de la demande sont ajustés en conséquence.

Si à la fois une ligne et une colonne sont satisfaites simultanément, une seule est décalée, la même que dans la méthode du Coin Nord-Ouest, Ensuite, recherchez la case non décalée avec le coût unitaire le plus petit et répétez le processus jusqu'à ce qu'une ligne où une colonne exactement soit laissée hors traitement.

➢ <u>les étapes de la méthode du Coût Minimum :</u>

1. Identifiez la case ayant un coût minimum de transport unitaire (C_{ij}).
2. S'il y a deux où plus de coûts minimaux, sélectionnez la ligne et la colonne correspondant à la rangée numérotée inférieure.
3. S'ils apparaissent dans la même rangée, sélectionnez la colonne numérotée inférieure.
4. Choisissez la valeur du x_{ij} correspondante autant que possible sous réserve des contraintes de disponibilité et de demande.
5. Si la demande est satisfaite, supprimez la colonne.
6. Si la disponibilité est épuisée, supprimez la ligne.
7. Répétez les étapes 1 à 6 jusqu'à ce que toutes les restrictions soient satisfaites.

➢ <u>Résumé des étapes de la méthode du Coût Minimum :</u>

1. Allouer autant que possible à la case avec le coût minimum de transport, et ajuster les demandes et les disponibilités.
2. Répétez l'étape 1 jusqu'à ce que toutes les contraints soient respectées.

2.3. Méthode d'Approximation de Vogel :

➢ <u>Définition et principe:</u>

La Méthode d'Approximation de Vogel (VAM): est basé sur la notion de minimisation des coûts d'opportunité (où de pénalité).
Le coût d'opportunité d'une colonne d'approvisionnement donnée où d'une colonne de demande est défini comme la différence entre le coût le plus bas et l'alternative suivante de coût le plus bas. Cette méthode est préférée par rapport aux méthodes discutées ci-dessus, car elle génère généralement une solution de démarrage optimale où proche de celle optimale.

Par conséquent, si nous utilisons la solution initiale obtenue par VAM et que nous procédons à la solution optimale, la quantité de temps nécessaire pour arriver à la solution optimale est considérablement réduite.

- Les étapes impliquées dans la détermination d'une solution de base initiale à l'aide de VAM sont les suivantes:

1. Écrivez le problème de transport donné sous forme de tableau (s'il n'est pas donné).
2. Calculez la différence entre le coût minimum et le coût minimum suivant correspondant à chaque ligne et chaque colonne qui est appelée coût de pénalité.
3. Choisissez la différence maximale où le coût de pénalité le plus élevé. Supposons qu'il corresponde à la i-ème rangée (colonne où ligne), Choisissez la case avec un coût minimum dans la rangée, Encore une fois, si le maximum correspond à une colonne, choisissez la case avec le coût minimum dans cette colonne.
4. Supposons que c'est la case (i, j). Affecter min (a_i, b_j) à cette case, Si le min (a_i, b_j) = a_i, alors la disponibilité de la i-ème origine est épuisée et la demande à la j- ème destination reste comme $b_j - a_i$ et la i-ème rangée est supprimée de la table. Encore une fois si min (a_i, b_j) = b_j, puis demande au j- ème La destination est remplie et la disponibilité à la i-ème origine reste pour être $a_i - b_j$ et la colonne j est supprimée de la table.
5. Répétez les étapes 2, 3, 4 avec le tableau restant jusqu'à ce que toutes les origines soient épuisées et que toutes les demandes soient satisfaits.

- Remarques :

- La Méthode d'Approximation de Vogel (VAM) est connue aussi par :l'heuristique de Balas-Hammer /la Méthode de la différence maximale / la méthode de pénalité unitaire /la méthode des regrets maximaux successifs.
- La méthode VAM est basée sur la notion de coût de pénalité où de regret.
- Un coût de pénalité est la différence entre le coût de case le plus grand et le plus important d'une rangée (ligne où colonne).
- Dans VAM, la première étape consiste à développer un coût de pénalité pour chaque source et destination.
- Le coût de pénalité est calculé en soustrayant le coût de case minimum du coût de case supérieur suivant dans chaque rangée.

- Résumé des étapes de la Méthode d'Approximation de Vogel (VAM):

1. Déterminer le coût de pénalité pour chaque ligne et chaque colonne.
2. Sélectionnez la ligne où la colonne avec le coût de pénalité le plus élevé.
3. Allouer autant que possible à la cellule faisable avec le coût de transport le plus bas dans la rangée (la ligne où la colonne) avec le coût de pénalité le plus élevé.
4. Répétez l'étape 1, 2 et 3 jusqu'à ce que toutes les contraintes soient respectées.

3. Méthodes d'optimisation de solution de base :

Pour obtenir une solution optimale en apportant des améliorations successives à la solution de base initiale jusqu'à ce qu'il n'y ait plus de réduction du coût de transport. Une solution optimale est celle où il n'y a pas d'autre ensemble de routes de transport qui réduira encore le coût total du transport.

Ainsi, nous devons évaluer chaque case inoccupée dans le tableau de transport sur l'opportunité de réduire le coût total du transport. Une case inoccupée avec le coût d'opportunité négatif le plus important est choisi pour inclure dans le nouvel ensemble de routes de transport (allocations).

Cette valeur indique la réduction de coût par unité qui peut être obtenue en augmentant l'allocation d'expédition dans la case inoccupée à partir de son niveau actuel de zéro. Ceci est également connu comme une case entrante (variable d'entré comme dans le simplexe). La case sortante (variable sortante) dans la solution actuelle est la case occupée (variable de base) dans le chemin fermé unique (boucle) dont l'attribution deviendra nulle, car d'autres unités sont attribuées à la case inoccupée avec le plus grand coût d'opportunité négatif.
C'est-à-dire que la solution actuelle ne peut plus être améliorée. C'est la solution optimale.

- Les méthodes largement utilisées pour trouver une solution optimale sont:

- Méthode Stepping stone.
- Méthode de distribution modifiée.

Elles diffèrent dans leurs mécanismes, mais leurs itérations sont les mêmes, ils donnent exactement les mêmes résultats (puisqu'elles sont des méthodes exactes) et utilisent la même stratégie de test. Pour développer la solution améliorée, si elle n'est pas optimale.

- Remarque :

Bien que le problème du transport puisse être résolu en utilisant la méthode du simplexe ordinaire, ses propriétés spéciales constituent une méthode plus pratique pour résoudre ce type de problèmes. Ces méthodes sont basées sur la même théorie de la méthode du simplexe. Cependant, elles utilisent des raccourcis qui fournissent un schéma de calcul moins lourd. Il existe une différence entre les deux méthodes et La méthode simplex, cette dernière effectue les opérations sur une table du simplexe. La méthode de transport effectue les mêmes opérations sur une table de transport.

3.1. Méthode de Stepping-Stone :

- Définition :

L'algorithme du Stepping-Stone est un algorithme itératif (donc par étapes successives) vise à améliorer une solution de base. (Faire baisser le coût global)

On peut appliquer l'algorithme à n'importe quelle solution de base.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

- <u>Déroulement de l'algorithme :</u>

L'algorithme consiste à modifier la solution pour une qui soit meilleure, donc à rendre non vide une case vide du tableau de transport.

- <u>Algorithme de stepping stone:</u>

1. Déterminer une solution de base initiale.

2. Calculer les coûts marginaux [1] $\delta_{ij} = c_{ij} - (t_j - t_i)$ avec $t_j - t_i = \theta_{ij}$ la tension de l'arc (i, j), t_i s'appelle le potentiel du sommet i de l'arc (i, j). Si tous les coûts marginaux sont positifs où nuls alors FIN. La solution est optimal, sinon passer à 3.

3. Pour tous les coûts marginaux négatifs, chercher la chaine de substitution et déterminer la quantité maximale qui peut être déplacé et passer à (4).Alors le gain correspondant est égale au produit de cette quantité par le coût marginale.

4. Retenir la substitution qui réalise la plus grande diminution du coût de transport, l'effectuer et revenir à (2).

- <u>Comment déterminer les potentiels ?</u>

- On utilisera le tableau des coûts limité aux cases où la quantité transitée est non nulle.
- On déterminera les potentiels de proche en proche : on commencera par une destination, puis une origine, puis une destination.

- <u>Comment déterminer les δ ?</u>

- Pour chaque case nulle, on calculera δ en ajoutant au coût unitaire de la case le potentiel d'origine associée et en retranchant le potentiel de la destination correspondante.

- <u>Comment déterminer les quantités à transporter(q) ?</u>

- On déterminera les quantités qu'on peut ajouter aux cases vides uniquement pour celles dont le δ est négatif, il ne sert à rien, en effet, de remplir une case qui fait augmenter le coût.
- Pour remplir une case vide, il faut diminuer une case pleine, donc constituer un circuit de cases pleines qu'on vide et remplit alternativement.

[1] Le coût marginale : la quantité (positive ou négative) qui s'ajout au coût globale lorsqu'on veut transporter une unité sur un arc de flux nul.

- Résumé d'heuristique de Stepping-Stone:

1. Déterminez les chemins d'accès et les changements de coûts pour chaque case vide dans le tableau.
2. Allouer autant que possible à la case vide avec la plus forte diminution nette des coûts.
3. Répétez l'étape 1 et 2 jusqu'à ce que toutes les cases vides aient des changements de coûts positifs qui indiquent une solution optimale.

3.2. Méthode de Distribution Modifiée :

- Définition :

La Méthode de distribution modifiée (où des penalties) : est une version modifiée de la méthode de stepping stone dans laquelle les équations mathématiques remplacent les chaines de substitutions. Cette méthode est plus pratique que stepping stone.

En appliquant la méthode MODI, nous commençons par une solution initiale obtenue en utilisant les méthodes citées à la section précédente. Ensuite, nous devons calculer une valeur u_i pour chaque ligne i et v_j pour chaque colonne j dans la table de transport.

- Les étapes de la méthode de distribution modifiée

1. Pour calculer les valeurs u_i et v_j pour chaque ligne et chaque colonne, définissez les équations : $u_i + v_j = c_{ij}$.

2. Une fois que toutes les équations ont été écrites, définissez l'une des deux variables u_i ou v_j à zéro, et résolvez le système d'équations pour toutes les valeurs u_i et v_j.

3. Calculez l'indice d'amélioration Δ_{ij} pour chaque cellule inutilisé par l'amélioration de la formule : $\Delta_{ij} = c_{ij} - u_i - v_j$.

4. Transférer la plus grande quantité possible à la cellule qui a Δ_{ij} le plus négatif en créant un cycle qui satisfait la demande et la disponibilité de chaque rangé.

5. Répétez les étapes 2 à 4 jusqu'à ce qu'il n'y ait pas de Δ_{ij} négatif.

6. Calculez le coût total en multipliant chaque allocation (x_{ij}) par son spécifique coût (c_{ij}).

- <u>Résumé des étapes de La méthode de distribution modifiée (MODI) :</u>

1. Développer une solution initiale.
2. Calculez les valeurs u_i et v_j pour chaque ligne et chaque colonne.
3. Calculer l'indice d'amélioration Δ_{ij} , pour chaque case vide.
4. Affecter autant que possible à la case vide qui entraînera la plus forte diminution du coût (Δ_{ij} le plus négatif). Répétez les étapes 2 à 4 jusqu'à ce que toutes les valeurs Δ_{ij}, soient positives où nulles.

Chapitre 4

Applications

Introduction :

Le modèle de transport est un modèle linéaire, et peut être donc traité par l'algorithme du simplexe. Cependant, la structure particulière de ce modèle permet de simplifier considérablement l'algorithme. Nous illustrons ici ce que devient l’algorithme lorsqu’on utilise les méthodes d’obtenir et d’optimiser une solution de base de problème de transport indiquées à la section précédente, en les appliquant à un exemple illustratif pour bien expliquer leurs fonctionnements, et manipulant l’algorithme sous forme d’un programme informatique (programmé en langage C) capable de résoudre un problème de transport équilibré.

1. L’exemple illustratif : (le transport des céréales : Le cas du blé.)

- <u>Le contexte :</u>

La France est dans le top 5 mondiaux de la production de blé. Elle est fortement exportatrice avec 70% de ses exportations qui vont vers les pays hors UE (notamment le Maghreb et l’Afrique subsaharienne). Le blé est à l’origine de toute une filière qui produit une gamme très vaste de produits qui va de la farine aux pains de la boulangerie. Aujourd’hui, un moulin, peut, selon les cas, tourner 24 heures sur 24, 365 jours par an où sur un rythme plus traditionnel pour produire la farine qui va ensuite être transformée sur d’autres sites industriels.

Figure 14: les sites industriels

Ce rythme, bien entendu, s’emballe lors des récoltes de juillet à août. Cette période voit une noria de camions s’agiter pour évacuer les céréales vers les silos de stockage des coopératives.

Ce pic d'activité pose un vrai défi à la logistique qui doit être suffisamment robuste pour l'absorber. Or la question est d'autant plus sensible, que cette période est celle des congés pour de nombreux transporteurs…

Pour donner un exemple, un moulin de taille familiale d'une dizaine de salariés peut être approvisionné par plus de 130 agriculteurs des environs (sources : La Voix du Nord). Ce sont autant de tournées de collecte à organiser.

Le transport de blé implique des matériels et des qualifications spécifiques selon les phases de sa transformation. Celle qui va du champ de l'agriculteur au moulin de la coopérative et celle qui va du moulin aux usines de transformation des industriels.

Figure_9: le transport du blé.

Des entreprises innovantes, comme viennent apporter de nouvelles réponses crédibles aux problématiques transport du blé. Peut ainsi proposer des transporteurs certifiés et notés qui connaît les spécificités du transport de céréales avec ses transporteurs audités afin de répondre au mieux au cahier des charges des chargeurs. Au bon produit, le bon transporteur. Enfin, sa solution est un outil de pilotage de flux qui permet d'optimiser les opérations des professionnels en employant des capacités de transport sous-exploitées ou vacantes, ce qui peut répondre à la problématique du transport des récoltes.

- <u>La problématique :</u>

Le transport de blé implique des matériels et des qualifications spécifiques selon les phases de sa transformation. Celle qui va du champ de l'agriculteur au moulin de la coopérative et celle qui va du moulin aux usines de transformation des industriels.

Le problème est de déterminer combien de tonnes de blé à transporter de chaque élévateur à grain à chaque moulin, afin de minimiser le coût total du transport.

2. Modélisation :

la variable de décision x_{ij} représente le nombre de tonnes de blé transporté par chaque élévateur i (où i = 1,2,3), à chaque moulin j (où j = A, B, C). La fonction objective représente le coût de transport total pour chaque itinéraire, Chaque terme dans la fonction objective reflète le coût du tonnage transporté pour une route.

Tableau 2.0: un problème de transport équilibré

	A	B	C	Disponibilités
1	6	8	10	150
2	7	11	11	175
3	4	5	12	275
Demandes	200	100	300	600

3. Détermination d'une solution de base initiale :

3.1. Méthode de Coin Nord-Ouest :

Dans la méthode du coin nord-ouest, la plus grande répartition possible est faite à la case dans le coin supérieur gauche du tableau, suivie d'allocations vers des cases adjacentes.

Nous attribuons d'abord autant que possible à la case 1A (coin nord-ouest). Ce montant est de 150 tonnes, puisqu'il s'agit du maximum pouvant être fourni par l'élévateur à grain 1, Même si 200 tonnes sont demandées par l'usine A. dans cette allocation initiale, est présentée dans le tableau 2.0. Nous allons ensuite allouer à une case adjacente à la case 1A, dans ce cas soit la case 2A où la case 1B.

Cependant, la case 1B ne représente plus une répartition possible, car le tonnage total de blé disponible à la source 1 (150 tonnes) a été alloué. Ainsi, la case 2A représente la seule alternative possible, et autant que possible est alloué à cette case, Le montant alloué à 2A peut être de 175 tonnes, la fourniture disponible à partir de la source 2 est 50 tonnes, le montant actuellement demandé à la destination A, Parce que 50 tonnes sont le montant le plus contraint, il est attribué à la case 2A. Comme le montre le tableau 2.0 .

La troisième répartition est faite de la même manière que la deuxième répartition la seule case possible adjacente à la case 2A est la case 2B. Le plus qui peut être attribué soit 100 tonnes (le montant demandé à l'usine B) soit 125 tonnes (175 tonnes moins les 50 tonnes attribuées à la case 2A) Le montant le plus petit (le plus contraint), 100 tonnes, est attribué à la case 2B, comme le montre le tableau 2.0 .

La quatrième allocation est de 25 tonnes à la case 2C et la cinquième allocation est de 275 tonnes à la case 3C, toutes deux reprises dans le tableau 2.1 .

Tableau 2.1. Solution obtenue par la méthode de Coin Nord-Ouest :

	A	B	C	Disponibilités
1	150	6	8	10	150
2	50	7	100	11	25	11	175
3	4	5	275	12	275
Demandes	200	100	300	600

Les allocations effectuées par la méthode de coin nord-ouest produisent donc une solution de base réalisable, puisque (m + n - 1) = 3 +3 - 1 = 5, ce qui équivaut au nombre d'allocations réalisées.

Comme le nombre de cases occupées 5 est égal (3 + 3-1), la condition est satisfaite,
La solution initiale est complète lorsque toutes les exigences sont satisfaites.

La solution de départ (composée de 4 variables de base) est :

x_{2A}= 50 tonnes.
x_{2B} = 100 tonnes.
x_{2C} = 25 tonnes.
x_{2C} = 275 tonnes.

Le coût du transport est calculé en évaluant la fonction objective :

Z = 6×1A + 8×1B + 10×1C + 7×2A + 11×2B + 11×2C + 4×3A + 5×3B + 12×3C
= 6× (150) + 8× (0) + 10× (0) + 7× (50) + 11× (100) + 11× (25) + 4× (0) + 5× (0) +12×(275)
= 5925 $.

3.2. Méthode du Coût Minimum :

Tableau 2.2.première itération :

	A	B	C	Disponibilités
1	6	8	10	150
2	7	11	11	175
3	200 4	5	12	275
Demandes	200	100	300	600

La case 3A est correspondante à la case de coût minimal.

Tableau 2.3. La deuxième itération : affectation minimale des coûts des cases :

	A	B	C	Disponibilités
1	6	8	10	150
2	7	11	11	175
3	200 4	75 5	12	275
Demandes	200	100	300	600

Tableau 2.4: La solution de départ en utilisant la méthode de Coût Minimum :

	A	B	C	Disponibilités
1	6	25 8	10	150
2	7	11	175 11	175
3	200 4	75 5	12	275
Demandes	200	100	300	600

La solution de départ (composée de 4 variables de base) est :

x_{1B} = 25 tonnes.
x_{2C} = 175 tonnes.
x_{3A} = 200 tonnes.
x_{3B} = 75 tonnes.

Le coût du transport est calculé en évaluant la fonction objective :

$$
\begin{aligned}
Z &= 8 \times x_{1B} + 11\times x_{2C} + 4\times x_{3A} + 5\times x_{3B} \\
&= 8\times (25) + 11\times (175) + 4 \times (200) + 5\times (75) \\
&= 4550 \$.
\end{aligned}
$$

➢ Remarque :

- La solution de base initiale obtenue par la méthode de Minimum est de coût total egale à 4550 $.
- La méthode de Minimum fournira une solution avec un coût inférieur à celui de la solution de coin nord-ouest car elle considère le coût de transport dans le processus d'allocation.

3.3. Méthode d'approximation de Vogel :

Application du procédé d'approximation de Vogel à l'exemple illustratif :

Tableau 2.6: L'allocation initiale du VAM.

	A	B	C	Disponibilités	Différence ligne
1	6	8	10	150	2
2	7	11	11	175	4
3	4	5	12	275	1
Demandes	200	100	300	600	
Différence colonne	2	3	1		

La méthode d'approximation de Vogel alloue autant que possible à la case de coût minimum dans la ligne ou la colonne avec le plus grand coût de pénalité.

Tableau 2. 7: La deuxième affectation VAM :

	A	B	C	Disponibilités	Différence ligne
1	6	8	10	150	2
2	175 7	11	11	175	
3	4	5	12	275	1
Demandes	200	100	300	600	
Différence colonne	2	3	2		

Tableau 2.8: Troisième affectation VAM :

	A	B	C	Disponibilités	Différence ligne
1	6	8	10	150	4
2	175 7	11	11	175	
3	4	100 5	12	275	8
Demandes	200	100	300	600	
Différence colonne	2		2		

Coûts de pénalité récompensés après la troisième affectation.

Table 2.9: la Solution de base Initial obtenue par la méthode de VAM:

	A	B	C	Disponibilités
1	6	8	150 10	150
2	175 7	11	11	175
3	25 4	100 5	150 12	275
Demandes	200	100	300	600

La solution de départ (composée de 4 variables de base) est :

x_{1C}= 150 tonnes.

x_{2A} = 175 tonnes.

x_{3A} = 25 tonnes.

x_{3B} = 100 tonnes.

x_{3C} = 150 tonnes.

Le coût du transport est calculé en évaluant la fonction objective :

$Z = 10 \times x_{1C} + 7\times x_{2A} + 4\times x_{3A} + 5\times x_{3B}+12\times x_{3C}$

$= 8\times (25) + 11\times (175) + 4 \times (200) + 5\times (75) = 5\ 125$ \$.

4. Optimisation de solution de base initiale :

La solution initiale utilisée comme point de départ de ce problème est la solution obtenue par la méthode du Coût Minimum parce qu'elle avait le coût total minimum des trois méthodes utilisées.

4.1. Méthode de Steeping-stone :

La méthode stepping-stone détermine s'il existe une cellule sans allocation qui réduirait le coût s'il est utilisé

Le principe de la solution de base dans un problème de transport est de déterminer si une route de transport qui n'est pas actuellement utilisée (c'est-à-dire une cellule vide) entraînerait un coût total plus faible si elle était utilisée. Par exemple, le tableau 2.10 montre quatre cellules vides (1A, 2A, 2B, 3C) représentant des itinéraires inutilisés.

Tableau 2.10: la solution de base du Coût Minimum :

	A	B	C	Disponibilités
1	6	25 / 8	125 / 10	150
2	7	11	175 / 11	175
3	200 / 4	75 / 5	12	275
Demandes	200	100	300	600

Notre première étape dans la méthode de stepping stone est d'évaluer ces cellules vides pour voir si l'utilisation de l'un d'entre eux réduirait le coût total. Si nous trouvons une telle route, nous allons allouer autant que possible.
Tout d'abord, envisageons d'allouer une tonne de blé à la cellule 1A. Si une tonne est attribuée à la cellule 1A, le coût sera augmenté de 6 $. Cependant, nous augmentons l'offre dans la rangée 1 à 151 tonnes, comme le montre le tableau 2.11.

Tableau 2.11: l'allocation d'une tonne à partir de la case 1A

	A	B	C	Disponibilités	
1	6	25 8	125 10	150	151
2	7	11	175 11	175	
3	200 4	75 5	12	275	
Demandes	200	100	300	600	

Les contraintes du problème ne peuvent être dépassées et la réalisabilité doit être maintenue. Si nous ajoutons une tonne à la cellule 1A, nous devons soustraire une tonne d'une autre répartition le long de cette ligne. La case 1B est un candidat logique car il contient 25 tonnes. En soustrayant une tonne de la cellule 1B, nous avons maintenant 150 tonnes dans la rangée 1, et nous avons satisfait à nouveau la contrainte d'approvisionnement. Dans le même temps, soustraire une tonne de la cellule 1B a réduit le coût total de 8 $.

Cependant, en soustrayant une tonne de la cellule 1B, nous avons maintenant seulement 99 tonnes attribuées à la colonne B, où 100 tonnes sont demandées, comme le montre le tableau B-13. Pour compenser cette violation de contrainte, une tonne doit être ajoutée à une cellule qui dispose déjà d'une allocation. Puisque la cellule 3B a 75 tonnes, nous ajouterons une tonne à cette cellule, ce qui rétablit encore la contrainte de 100 tonnes.

Tableau 2.12: Soustraction d'une tonne à partir de la case 1B

	A	B	C	Disponibilités
1	+1 6	25 -1 8	125 10	150
2	7	11	175 11	175
3	200 4	75 5	12	275
Demandes	200	100	300	600
		99		

Il faut soustraire une tonne d'une autre répartition le long de cette ligne.

En allouant une tonne supplémentaire à la cellule 3B, nous avons augmenté le coût de 5 $, le coût de transport pour cette cellule. Cependant, nous avons également augmenté l'offre dans la rangée 3 à 276 tonnes, une violation de la contrainte d'approvisionnement pour cette source. Comme auparavant, cette violation peut être corrigée en soustrayant une tonne de la cellule 3A, qui contient une allocation de 200 tonnes.
Cela répond à nouveau à la contrainte d'approvisionnement pour la rangée 3 et réduit également le coût total de 4 $, le coût de transport pour la cellule 3A. Ces allocations et suppressions sont présentées dans le tableau 2.13 :

Tableau 2.13: l'ajout d'une tonne à la case 3B et la soustraction d'une tonne à partir de la case 3A

	A	B	C	Disponibilités
1	+1	6	25 -1	8	125	10	150
2	7	11	175	11	175
3	200 -1	4	75 +1	5	12	275
Demandes	200	100	300	600

Notez dans le tableau 2.13 qu'en soustrayant une tonne de la cellule 3A, nous n'avons pas violé la contrainte de la demande pour la colonne A, puisque nous avons précédemment ajouté une tonne à la cellule 1A.

Examinons maintenant les augmentations et les réductions des coûts résultant de ce processus.

1A→1B→3B→3A
+6 - 8 +5 – 4 = - 1 $

En d'autres termes, pour chaque tonne allouée à la cellule 1A (une route qui n'est pas actuellement utilisée), le coût total sera réduit de 1 $. Ceci indique que la solution initiale n'est pas optimale car un coût plus faible peut être obtenu en allouant des tonnes supplémentaires de blé à la cellule 1A (c'est-à-dire que la cellule 1A est analogue à une colonne pivot dans la méthode simplex).

Notre objectif est de déterminer la cellule ou d'entrer une «variable» qui réduira le coût : Une autre variable (cellule vide) peut entraîner une diminution encore plus importante du coût que la cellule 1A. Si une telle cellule existe, elle sera sélectionnée comme variable d'entré, Sinon, la cellule 1A sera sélectionnée. Pour identifier la variable de saisie appropriée, les cellules vides restantes doivent être testées à mesure que la cellule 1A.

Tableau 2.14: Le chemin de Stepping-Stone pour la case 2A :

	A	B	C	Disponibilités
1	6	25 - 8	125 + 10	150
2	+ 7	11	175 - 11	175
3	200 - 4	75 + 5	12	275
Demandes	200	100	300	600
Le chemin : 2A→2C→1C→1B→3B→3A→2A			Le coût : +7-11+10-8+5-4= -1	

Une case vide qui réduira le coût est une variable d'entrée potentielle.
Pour évaluer le potentiel de réduction des coûts d'une case vide, un cycle reliant les cases utilisées aux cases vides est identifié.

Les chemins de trame restants et les calculs résultants pour les cases 2B et 3C

Tableau 2.15: Le chemin de steeping stone pour la case 2B :

	A	B	C	Disponibilités
1	6	25 + 8	125 - 10	150
2	7	- 11	175 + 11	175
3	200 4	75 5	12	275
Demandes	200	100	300	600
Le chemin : 2B→2C→1C→1B→2B			Le coût : +11-11+10 - 8= + 2	

Tableau 2.16: Le chemin de steeping stone pour la case 3C :

	A	B	C	Disponibilités
1	6	25 + 8	125 - 10	150
2	7	11	175 11	175
3	200 4	75 - 5	+ 12	275
Demandes	200	100	300	600
Le chemin : 3B→3C→1C→			Le coût : + 12 - 10 + 8 - 5 = + 5	

Tableau 2.17: Le chemin de Le chemin de steeping stone pour la case 1A

	A	B	C	Disponibilités
1	+ 6	25 - 8	125 10	150
2	7	11	175 11	175
3	200 - 4	75 + 5	12	275
Demandes	200	100	300	600
Le chemin : 1A→3A→3B→1B→1A			Le coût : +6 - 4 +5 - 8 = -1	

Après que toutes les cellules vides sont évaluées, celle avec le plus grand potentiel de réduction des coûts est la variable d'entré.

Tableau 2.18: deuxième Itération de Stepping-Stone :

	A	B	C	Disponibilités
1	25 6	8	125 10	150
2	7	11	175 11	175
3	175 4	100 5	12	275
Demandes	200	100	300	600

Lors de la réaffectation d'unités à la variable d'entré (case), le montant est la quantité minimale soustraite sur le chemin d'accès.

À chaque itération, une variable entre et une part (comme dans la méthode simplex).

Tableau 2.19: Le chemin de steeping stone pour la case 2A

Vérifiez si la solution est optimale.

	A	B	C	Disponibilités
1	25 − 6	8	125 + 10	150
2	+ 7	11	175 − 11	175
3	175 4	100 5	12	275
Demandes	200	100	300	600
Le chemin : 2A→2C→1C→1A→2C			Le coût : +7 – 11 + 10 – 6 = 0	

Tableau 2.20: Le chemin de steeping stone pour la case 1B

	A	B	C	Disponibilités
1	25 − 6	+ 8	125 10	150
2	7	11	175 11	175
3	175 + 4	100 − 5	12	275
Demandes	200	100	300	600
Le chemin : 1B→1A→3A→3B→1B			Le coût : +8 – 6 + 4 – 5 = +1	

Tableau 2.21: Le chemin de steeping stone pour la case 2B

Vérification de l'optimalité :

	A	B	C	Disponibilités
1	25 − 6	8	125 + 10	150
2	7	+ 11	175 − 11	175
3	175 + 4	100 − 5	12	275
Demandes	200	100	300	600
Le chemin : 2B →3B→3A→1A→1C→2C→2B			Le coût : +11 – 5 +4 – 6 +10 – 11 = + 3	

Le processus de vérification est répété jusqu'à ce qu'aucune des cases vides ne réduise les coûts (c'est-à-dire une solution optimale).

Par exemple, l'évaluation de quatre chemins indique que aucune réduction de coût ; Par conséquent, la solution du Tableau 16 est optimale.

La solution :

x_{1A} = 25 tons,
x_{2C} = 175 tons,
x_{3A} = 175 tons,
x_{1C} = 125 tons,
x_{3B} = 100 tons

Z = 6×(25) + 8×(0) + 10×(125) + 7×(0) + 11× (0) + 11× (175) + 4× (175) + 5× (100) + 12× (0)
= 4525 $

Tableau2.23: la solution alternative optimale :

	A	B	C	Disponibilités
1	6	8	150 / 10	150
2	25 / 7	11	150 / 11	175
3	175 / 4	100 / 5	12	275
Demandes	200	100	300	600

➢ Remarque :

- Une solution optimale multiple se produit lorsqu'une case vide a un changement de coût de zéro et toutes les autres cases vides sont positives.

- Une solution optimale alternative est déterminée en allouant à la case vide avec un changement de coût nul.

- Le coût minimum total optimal est également égal à 4525 $.

4.2. Méthode de distribution modifiée :

Nous utiliserons à nouveau la solution de base initiale obtenue par la méthode du coût minimum. Le tableau pour la solution initiale avec les modifications requises par Methode de distribution modifiée est présenté dans le tableau 2.24 :

Tableau 2.24: La solution initiale de coût minimum

	$v_j =$	$v_A =$	$v_B =$	$v_C =$	
$u_i =$		**A**	**B**	**C**	Disponibilités
$u_1 =$	1	$\boxed{6}$	25 $\boxed{8}$	125 $\boxed{10}$	150
$u_2 =$	2	$\boxed{7}$	$\boxed{11}$	175 $\boxed{11}$	175
$u_3 =$	3	200 $\boxed{4}$	75 $\boxed{5}$	$\boxed{12}$	275
	Demandes	200	100	300	600

Dans la table, la colonne supplémentaire à gauche avec les symboles u_i et la ligne supérieure supplémentaire avec les symboles v_J représentent des valeurs qui doivent être calculées.

Calculer pour toutes les cellules allouées:
$u_i + v_j = c_{ij}$: Coût de transport unitaire pour la case ij.

Formules pour les cellules contenant des allocations:

$x_{1B} : u_1 + v_B = 8$
$x_{1C} : u_1 + v_C = 10$
$x_{2C} : u_2 + v_C = 11$
$x_{3A} : u_3 + v_A = 4$
$x_{3B} : u_3 + v_B = 5$

Maintenant, il existe cinq équations avec six inconnues. Pour résoudre ces équations, il est nécessaire d'assigner une seule des inconnues à une valeur nulle.

On met donc $u_1 = 0$ et on obtient:
$v_B = 8, v_C = 10,\ u_2 = 1\ , u_3 = -3, v_A = 7.$

Tableau 2.25: La solution initiale avec toutes les valeurs u_i et v_j

	v_j	$v_A = 7$	$v_B = 8$	$v_C = 10$	
u_i		A	B	C	Disponibilités
$u_1 = 0$	1	6	25 / 8	125 / 10	150
$u_2 = 1$	2	7	11	175 / 11	175
$u_3 = -3$	3	200 / 4	75 / 5	12	275
	Demandes	200	100	300	600

- Utilisez la formule des penalties pour évaluer toutes les cellules vides:

$$c_{ij} - u_i - v_j = \Delta_{ij}$$

Où Δ_{ij} est égal à l'augmentation où à la diminution (penalties) du coût qui se produirait en allouant à une cellule.

Pour les cellules vides du tableau 26:

$x_{1A}: \Delta_{1A} = c_{1A} - u_1 - v_A = 6 - 0 - 7 = -1$
$x_{2A}: \Delta_{2A} = c_{2A} - u_2 - v_A = 7 - 1 - 7 = -1$
$x_{2B}: \Delta_{2B} = c_{2B} - u_2 - v_B = 11 - 1 - 8 = +2$
$x_{3C}: \Delta_{3C} = c_{3C} - u_3 - v_C = 12 - (-3) - 10 = +5$

Ces calculs indiquent que la cellule 1A ou la cellule 2A diminuera le coût de 1 $ par tonne ciblée. Notez que ce sont exactement les mêmes changements de coût pour les quatre cellules vides, comme cela a été calculé dans la méthode de stepping stone. C'est-à-dire que la même information est obtenue en évaluant les chemins dans la méthode de stepping stone et en utilisant les formules mathématiques du Méthode de distribution modifiée.

Nous pouvons sélectionner soit la cellule 1A soit la 2A, car elles sont liées à 1. Si la cellule 1A est sélectionnée comme entrée de la variable hors base, il faut déterminer le chemin d'accès de cette cellule afin que nous sachions combien de réaffecter. La réaffectation selon ce chemin donne le tableau présenté dans le tableau 2.27

Tableau 2.27: La deuxième itération de la méthode :

Après chaque attribution à une cellule vide, les valeurs u_i et v_j doivent être recalculées

	$v_j =$	$v_A = 6$		$v_B = 7$		$v_C = 10$		
$u_i =$		**A**		**B**		**C**		Disponibilités
$u_1 = 0$	1	25	6		8	125	10	150
$u_2 = 1$	2		7		11	175	11	175
$u_3 = -2$	3	175	4	100	5		12	275
	Demandes	200		100		300		600

Les nouvelles valeurs u_i et v_j :

$x_{1A} : u_1 + v_A = 6$ Donc : $v_A = 6$
$x_{1C} : u_1 + v_C = 10$ Donc : $v_C = 10$
$x_{2C} : u_2 + v_C = 11$ Donc : $u_2 = 1$
$x_{3A} : u_3 + v_A = 4$ Donc : $u_3 = -2$
$x_{3B} : u_3 + v_B = 5$ Donc : $v_B = 7$

Changements de coûts pour les cellules vides : $c_{ij} - u_i - v_j = \Delta_{ij}$:

x_{1B}: $\Delta_{1B} = c_{1B} - u_1 - v_B = 8 - 0 - 7 = +1$
x_{2A}: $\Delta_{2A} = c_{2A} - u_2 - v_A = 7 - 1 - 6 = 0$
x_{2B}: $\Delta_{2B} = c_{2B} - u_2 - v_B = 11 - 1 - 7 = +3$
x_{3C}: $\Delta_{3C} = c_{3C} - u_3 - v_C = 12 - (-2) - 10 = +4$

Toutes les valeurs Δ_{ij} sont positives où nuls, donc la solution obtenue est optimale.

x_{1A} = 25 tons,
x_{2C} = 175 tons,
x_{3A} = 175 tons,
x_{1C} = 125 tons,
x_{3B} = 100 tons

Le coût total de transport est :
Z = 6×(25) + 8×(0) + 10×(125) + 7×(0) + 11× (0) + 11× (175) + 4× (175) + 5× (100) + 12× (0)
= 4525 $.

5. Implémentation en langage C :

5.1. Le langage C :

➢ Définition :

Le langage C a été mis au point par *D.Ritchie et B.W.Kernighan* [2] au début des années 70. Leur but était de permettre de développer un langage qui permettrait d'obtenir un système d'exploitation de type `UNIX` portable. D.Ritchie et B.W.Kernighan se sont inspirés des langages B et BCPL, pour créer un nouveau langage : le langage C.

➢ Caractéristiques :

- Le langage C reste un des langages les plus utilisés actuellement. Cela est dû au fait que le langage C est un langage qui comporte des instructions et des structures de haut niveau (contrairement à l'assembleur par exemple) tout en générant un code très rapide grâce à un compilateur très performant.

- Un des principaux intérêts du C est que c'est un langage très portable. Un programme écrit en C en respectant la norme ANSI[3] est portable sans modifications sur n'importe quel système d'exploitation disposant d'un compilateur C.
- La rapidité des programmes écrits en C est en grande partie due au fait que le compilateur présuppose que le programmeur sait ce qu'il fait : il génère un code ne contenant pas de vérifications sur la validité des pointeurs, l'espace d'adressage, etc. Ainsi, les programmes en C sont très compacts.

5.2. Logiciel de programmation :

Le logiciel ***Dev-C++*** est un environnement de développement intégré permettant de programmer en C/C++. Il utilise la version MinGW du compilateur GCC (venu du monde du logiciel libre) et permet d'exporter ses projets sous fichiers .dev.

Le compilateur ***Dev-C++*** est assez complet. Il comprend entre autre un "répertoire de classes", un "répertoire de fonctions incluses", et un débogueur qui permet de surveiller l'état des variables pendant l'exécution du programme. L'avantage de ***Dev-c++,*** c'est qu'il est multiplateforme. Il permet de générer des exécutables fonctionnels sous la majorité des plateformes. L'outil de développement ***Dev-c++*** peut être très pratique. Il est important de savoir s'en servir efficacement.

Figure_11 : Le compilateur Dev-C++

[2] Dennis MacAlistair Ritchie, (1941-2011) , Brian Kernighan (1942-…) sont des pionniers de l'informatique moderne, inventeur du langage C et codéveloppeur de Unix. sont des informaticiens connu pour avoir coécrit le premier livre sur le langage de programmation C

[3] L'American National Standards Institute (ANSI, Institut de normalisation américaine) est un organisme privé à but non lucratif qui supervise le développement de normes pour les produits, les services, les procédés, les systèmes et les employés des États-Unis

5.3. Programme de résolution de problème de transport :

- Programmation en langage C de l'algorithme de résolution de problème de transport :

Le code source du programme est attaché dans le CD : PFE-PT_M-IKEN

- L'exécution de programme :

```
C:\Users\DELL\Desktop\IKEN_Med\PROJET_PFE\APPLICATIONS\SOLUTION_DE_BASE.exe

PROBLEME DE TRANSPORT

NOMBRE DE SOURCES ? 3

NOMBRE DE DESTINATIONS ? 3

LA SAISIE DE COUTS DE TRANSPORTS :
 DE SOURCE #1 AU DESTINATION #1 ? 6
 DE SOURCE #1 AU DESTINATION #2 ? 8
 DE SOURCE #1 AU DESTINATION #3 ? 10

 DE SOURCE #2 AU DESTINATION #1 ? 7
 DE SOURCE #2 AU DESTINATION #2 ? 11
 DE SOURCE #2 AU DESTINATION #3 ? 11

 DE SOURCE #3 AU DESTINATION #1 ? 4
 DE SOURCE #3 AU DESTINATION #2 ? 5
 DE SOURCE #3 AU DESTINATION #3 ? 12


LA SAISIE DES DISPONIBILITES EXISTANT AUX SOURCES:
La Disponibilite de SOURCE #1 ? 150
La Disponibilite de SOURCE #2 ? 175
La Disponibilite de SOURCE #3 ? 275

LA SAISIE DES DEMANDEES EXISTANT AUX DESTINATIONS:
La Demande de DESTINATION #1 ? 200
La Demande de DESTINATION #2 ? 100
La Demande de DESTINATION #3 ? 300

CE PROBLEME EST EQUILIBRES

CHOISIR LA METHODE DE DETERMINER LA SOLUTION DE BASE
POUR LA METHODE DE COIN NORD OUEST ? VEULLER TAPER 1

POUR LA METHODE DE COUT MINIMUM ? VEULLER TAPER 2

POUR LA METHODE D'APPROXIMATION DE VOGEL ? VEULLER TAPER 3
1

LE COUT TOTAL DE TRANSPORT PAR LA METHODE DE NORD-OUEST EST Z= 5925

nb ieration: 5 iterations

CHOISIR LA METHODE DE DETERMINER LA SOLUTION DE BASE
POUR LA METHODE DE COIN NORD OUEST ? VEULLER TAPER 1

POUR LA METHODE DE COUT MINIMUM ? VEULLER TAPER 2

POUR LA METHODE D'APPROXIMATION DE VOGEL ? VEULLER TAPER 3
2
LE COUTS TOTAL DE TRANSPORT PAR LE COUT MINMUM EST  Z= 4550

nb ieration: 3 iterations
```

implémentation des données

le problème est équilibrée

Le programme vous demande de choisir la méthode pour déterminer la solution de base parmi les trois méthodes.

Le coût de transport : z=5925.
Déterminé en 5 itérations.

Le coût de transport : z=4550.
Déterminé en 3 itérations.

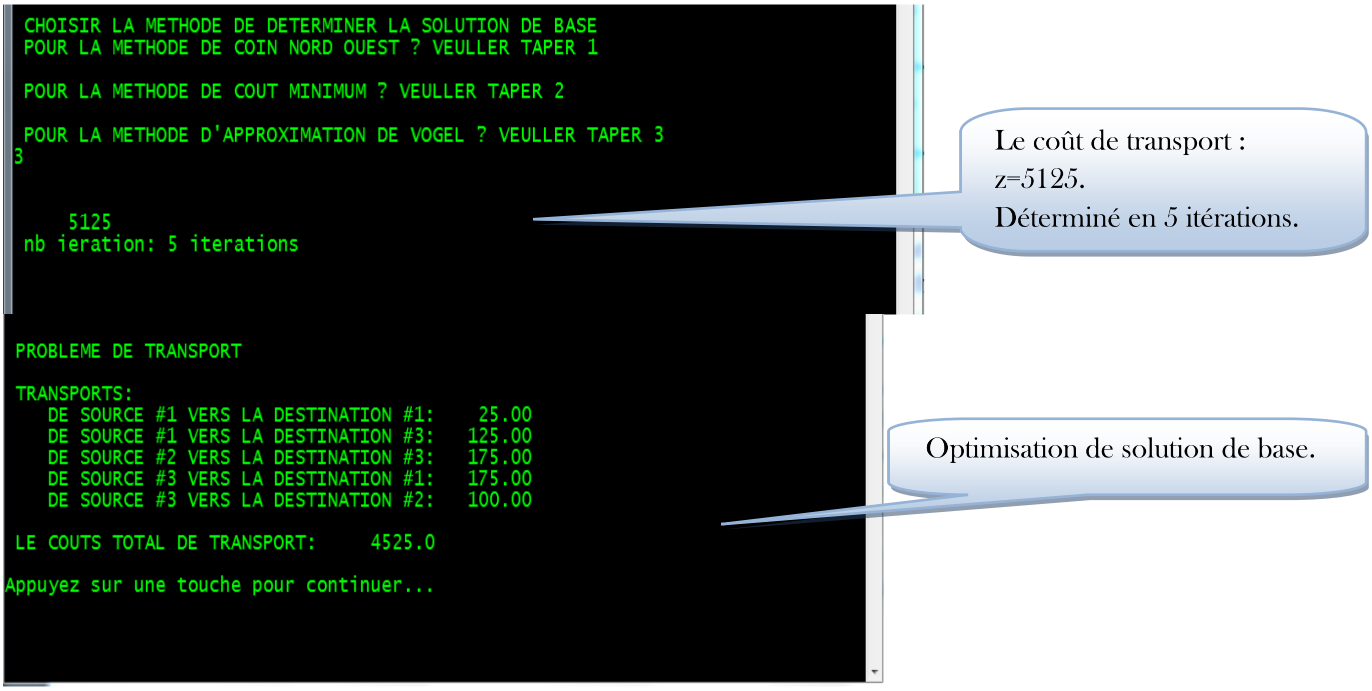


Figure_12: Exécution de programme

6. Comparaison des Méthodes :

Après la résolution de dix problèmes de transport de différentes tailles par toutes les méthodes qu'on a cité dans la section précédente (résolution d'un problème de transport), on a résumé les résultats des coûts de transport (en dollar $) dans le tableau suivant :

Problème :	Taille de problème	MCNO	MCM	MAV	Optimisation
1	3x4	3860	3670	3520	3460
2	3x3	6600	6460	5920	5920
3	3x3	730	555	555	555
4	3x5	363	305	290	290
5	3x4	176	150	149	149
6	3x3	5925	4550	4525	4525
7	3x4	7750	7650	7350	7225
8	3x4	310	180	180	180
9	3x4	670	650	610	610
10	3x3	175	148	150	148

- Analyse de données :

Nous avons utilisé ici quatre méthodes : Méthode de coin Nord-ouest (MCNO), méthode de coût Minimum (MCM), la méthode d'approximation de Vogel (VAM) pour trouver une première solution possible de base pour le modèle de transport. Et la méthode d'optimisation de la solution de base (steeping stone où méthode de distribution modifiée).
Le coût du transport montre que:

- Dans (VAM), la majorité des résultats sont meilleurs (optimisation) que les deux méthodes MCNO et MCM.

- A titre d'exemple le problème de transport numéro 1 est de taille 3x4 (c.-à-d. le nombre de sources 3 et le nombre de destinations est 4), ce problème est équilibré. Une solution de base obtenu par la méthode de Coin Nord-Ouest est de coût total de transport 3860 $, Or avec la méthode de coût minimum, la solution de base obtenu est de coût total inferieur à celui de Nord-ouest égale à 3670, restant dans le même problème la méthode d'approximation de Vogel donne une solution de base de coût total de transport 3520$ est le plus proche d'optimum et la solution optimale est égale 3460.
- Ce qui signifie que l'utilisation de la méthode d'approximation de Vogel pour la détermination de solution de base initiale réduit les calcules d'optimiser la solution de base initiale dans un temps plus optimal, et même dans certain problème la méthode d'approximation donne directement la solution optimale comme le montre le problème numéro 2,numéro 4 (de taille 3x3 ,3x5) respectivement la solution de base obtenu par VAM est identique à celle d'optimale.
- la méthode de Minimum est meilleure que de coin nord-ouest et dans certains problèmes, le résultat de MCM est meilleur que VAM (le problème 10).

➢ <u>Conclusion:</u>

Le concept de problèmes de transport équilibrés et les méthodes existantes disponibles pour résoudre ce type de problème de programmation linéaire dans leur ordre d'efficacité ont été examinés. Nous avons utilisé notre modification ainsi que d'autres méthodes existantes pour résoudre un problème de transport équilibré sélectionné; Et ont comparé l'efficacité de toutes les méthodes pour réduire le coût total en utilisant le même problème sélectionné. En suivant ce qui précède, il est évident que la méthode d'approximation Vogel présente un meilleur état que ses prédécesseurs en termes de réduction du coût total.

Conclusion générale:

Cette étude m'a donné l'opportunité de me familiariser au domaine de la recherche opérationnelle, ce domaine qui est la discipline des méthodes scientifiques pour aider à mieux décider et traiter les problèmes stratégiques et économiques, Le problème de transport est l'un de ces problèmes classique les plus connus, Mais la complexité et la variation des contraintes de ce problème dans le domaine économique impliquent la recherche d'autres heuristiques et même des métaheuristiques plus efficaces pour la résolution. Ce qui rend difficile de tirer une conclusion définitive sur la résolution de ce type des problèmes.

Dans ce rapport, on s'est intéressé d'avantage à la modélisation et la résolution de problème de transport équilibré par des différentes méthodes qui nous permettons d'obtenir une solution de base réalisable (Nord-Ouest, Coût minimum, Approximation de vogel), Ensuite nous avons essayé d'expliquer l'optimisation d'un telle solution de base initiale par la méthode de stepping stone et la méthode de distribution modifiée, puis nous avons essayé de faire une comparaison entre ces méthodes et les programmer en langage C comme un programme de résolution d'un problème de transport.

A la fin je peux dire que ce travaille représente une base de départ pour résoudre les problèmes générale de transport.

Bibliographie:

- [1] Mr. ETTAOUIL, Professeur à la Faculté des Sciences et Technique de Fès, Cours de la recherche opérationnelle année 2015/2016.
- [2] Programmation Mathématique (Pr. Ahmed El Hilali Alaoui, Pr Youssef Benadada).
- [3] Méthodes de planification en transport (Yves Nobert, Roch Ouellet, Régis Parent).
- [4] Mathematical Theory and Modeling ISSN 2224-5804 (Paper) ISSN 2225-0522 (Online) Vol.5, No.4, 2015
- [5] Optimisation appliquée (Yadolah Dodge).
- [6] Article: ALFRED ASASE. The transportation problem. Theses submitted to the department of mathematics faculty of physical science and technology Kumasi, October 2011, pages 19-53.

- http://books.google.fr/
- http://www.iiste.org/book/
- http://www.doc-etudiant.fr/
- http://www.iecn.unancy.fr/~garet/cours/graphes/graphes.pdf
- http://cours.ensem.inpl-nancy.fr/coursdm/graphes/glossary.html