République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 20 Août 1955 Skikda

Faculté des Sciences - Département d'informatique

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Spécialité : Génie Logiciel Avancé et Application

Thème

Problème de Transport

Réalisé par :

- ***Balaska Oussama***
- ***Larid Chouaib***

Encadré par :

- Dr Mallem.Z

Juin 2022

Dédicace

A nos chers parents, en témoignage de notre gratitude, si grande qu'elle puisse être, pour tous les sacrifices qu'ils ont consentis pour notre bienêtre et le soutien qu'ils nous ont prodigués tout le long de notre éducation.

Que Dieu, le tout puissant, les préserve et leur procure santé et longue vie.

A nos chers frères et sœurs pour le soutien et l'encouragement qu'ils nous ont accordé et à nos amis pour leur compréhension et fidélité, nous exprimons nos profonds remerciements.

Notre reconnaissance et notre grand respect s'adressent à nos professeurs pour leurs efforts remarquables.

Nous dédions ce travail, expression de notre grand amour avec tous nos vœux de bonheur et de prospérité.

Remerciement

Louange A Dieu, le miséricordieux, sans Lui rien de tout cela n'aurait pu être.

Nous tenons tout d'abord à remercier Dr Mallem.Z, pour l'honneur qu'il nous a fait en acceptant de nous encadrer. Ces conseils précieux ont permis une bonne orientation dans la réalisation de ce modeste travail.

Nous tenons à exprimer notre profonde gratitude à l'ensemble du corps enseignant qui a contribué à notre formation.

Enfin nous tenons à rendre hommage à toutes nos familles et nos amis pour le soutien qu'ils nous ont apportés durant toutes ces années d'études.

Résume

L'objectif de ce travail est de modéliser et de résoudre le problème de transport équilibré (le problème de transport consiste à déterminer la façon optimale d'acheminer des biens à partir de m entrepôts et de les transporter vers n destinations et cela à moindre cout) de différentes manières permettant d'obtenir une solution de base exploitable (Méthode de Coin Nord-Ouest, Méthode du Coût Minimum), puis d'améliorer cette solution de base initiale par la méthode de Stepping- Stone et la méthode de distribution modifiée, puis programmez-la dans un langage Java.

Mots clé : optimisation combinatoire, recherche opérationnelle, programmation linéaire.

Abstract

The objective of this work is to model and solve the balanced transport problem (the transport problem consists in determining the optimal way of transporting goods from m warehouses and transporting them to n destinations at the lowest cost) in different ways to obtain a workable basic solution (North-West Corner Method, Minimum Cost Method), then to improve this initial basic solution by the Stepping-Stone method and the modified distribution method, then program it in a Java language.

keywords: combinatorial optimization, operations research, linear programming.

الملخص

الهدف من هذا العمل هو نمذجة وحل مشكلة النقل المتوازن (تتمثل مشكلة النقل في تحديد الطريقة المثلى لنقل البضائع من مستودعات m ونقلها إلى عدد n من الوجهات بأقل تكلفة) بطرق مختلفة للحصول على حل أساسي عملي (طريقة الزاوية الشمالية الغربية، طريقة التكلفة الدنيا)، ثم لتحسين هذا الحل الأساسي الأولي من خلال طريقة -Stone Stepping- وطريقة التوزيع المعدلة، ثم برمجته بلغة Java.

الكلمات الرئيسية: التحسين التوافقي، بحوث العمليات، البرمجة الخطية.

TABLE DES MATIERE

TABLE DES FIGURES

Introduction Générale

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 par son envergure, créa un 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.

L'importance de l'optimisation est la nécessité d'utiliser des outils simples pour la modélisation Questions d'ordre économique, militaire ou autre prise de décision, la programmation linéaire est devenue l'un des domaines de recherche les plus actifs dans ce domaine. Les premières œuvres (1947) étaient des œuvres de George B. Danzg et son personnel de l'US Air Force Surtout pour résoudre certains problèmes, comme une implantation optimale Gestion optimale du radar de surveillance ou de la flotte d'approvisionnement.

La programmation linéaire implique généralement l'allocation de ressources limitées le plus possible pour maximiser ou minimiser les profits coût. Ces décisions sont généralement le résultat de problèmes mathématiques. En fabrication, la recherche opérationnelle peut Mieux organiser le plan de production.

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, nous énumérons quelques notions de base sur la programmation linéaire et une des méthodes de résolution d'un programme linéaire qui est la méthode du simplexe.

- 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 Java, les résultats obtenus, nous terminons le manuscrite par une conclusion générale et quelques perspectives.

Chapitre 1

La programmation linéaire

Chapitre 1

La programmation linéaire

Introduction

En optimisation combinatoire, deux types de représentations peuvent être suivies pour la modélisation des problèmes de transport avant de les résoudre : une représentation sous forme d'un graphe ou une autre représentation sous forme d'un programme linéaire.

Dans ce chapitre, nous rappelons d'abord certaines notions sur la programmation linéaire. Ensuite, nous abordons la méthode de résolution de problème de programmation linéaire.

1. La programmation linéaire

1.1 Généralités et définitions

La programmation linéaire traite de manière générale d'un problème d'allocation de ressources limitées parmi des activités concurrentes et ce d'une façon optimale.

La programmation linéaire emploie un modèle mathématique qui décrit le problème réel.

L'adjectif "linéaire" indique que toutes les fonctions mathématiques de ce modèle sont linéaires tandis que le terme "programmation" signifie essentiellement planification.

Définition 1.1

Soient D un domaine de $\mathbb{R}^n$ et une fonction f : D→ $\mathbb{R}$.

Un programme mathématique, noté (*P*), est le problème qui consiste à chercher un élément x^* de D tel que $f(x^*)$ soit le plus grand possible en cas de maximisation (resp. Le plus petit possible en cas de minimisation). C'est-à-dire [1]

$$\forall x \in D : f(x) \leq f(x^*) \text{ en cas de maximisation}$$

$$(\text{resp. } \forall x \in D : f(x) \geq f(x^*) \text{ en cas de minimisation})$$

Notation : Un programme mathématique (P) est représenté par :

$$\max_{x \in D} f(x) \quad problème\ de\ maximisation$$

$$\min_{x \in D} f(x) \quad problème\ de\ minimisation$$

Définition 1.2 *Soit* (*P*) *un programme mathématique.* [1]

- *Un point* $x \in D$ *est appelé* "***solution réalisable***" *de* (*P*).
- *Le domaine* D *est appelé* "***domaine des solutions réalisables***" *de* (*P*).
- *Une solution* x^* *de* (*P*)*, lorsqu'elle existe, est appelée* "***solution optimale***" *de* (*P*).
- *La fonction f est appelée* "***fonction objectif*** " *de* (*P*).
- $f\ (x^*)$ *, si* x^* *existe, est appelée* "***valeur du programme mathématique***" (*P*).

1.2 Formulation de problèmes de PL

1.2.1 Les étapes de formulation d'un PL

- ✓ La première étape dans la formulation d'un problème de programmation linéaire est de déterminer quelle quantité t que vous devez connaître pour résoudre le problème. On les appelle les **variables de décision**.

  -On note x_j la variable de décision avec j = 1, …, n

- ✓ La deuxième étape consiste à déterminer quelles sont les **contraintes** du problème. Par exemple, il peut y avoir une limite de ressources ou une valeur maximale ou minimale une variable de décision peut prendre, ou il pourrait y avoir une relation entre deux variables de décision.

$$\sum_{j=1}^{n} a_{ij}x_j = \leq, =, \geq b_i \qquad i = 1, \ldots, m$$

  -les nombre a_{ij} et b_i des constantes réelles et m le nombre des contraintes.

- ✓ La troisième étape consiste à déterminer l'objectif à atteindre. C'est la quantité à être maximisé ou minimisé, c'est-à-dire optimisé. La fonction des variables de décision qui à optimiser est appelée **la fonction objectif**.

$$z = \sum_{j=1}^{n} c_j x_j = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

-c_j : c'est le coefficient de contribution de la variable $x_j$dans la fonction objectif.

1.2.2 Exemple de modèle linéaire

Exemple 1.1 : Une société produit de la peinture d'intérieur et d'extérieur à partir de deux produits de base M1 et M2

Données :

	Quantité utilisée par tonne		Quantité disponible par jour
	Extérieure	Intérieure	
M1	6	4	24
M2	1	2	6
Profit par tonne	5	4	

- *Formulation :*

I. *Variables de décision :*

x_1 = tonnes de peinture d'extérieur produites par jour

x_2 = tonnes de peinture d'intérieur produites par jour

II. *Fonction objectif à optimiser :*

Max z = $5x_1 + 4x_2$

III. *Contraintes*

$$
\begin{aligned}
6x_1 + 4x_2 &\leq 24 \\
1x_1 + 2x_2 &\leq 6 \\
x_2 &\leq 2 \\
x_2 - x_1 &\leq 1 \\
x_1, x_2 &\geq 0
\end{aligned}
$$

1.3 Différentes formes d'un PL

Il est d'usage de considérer qu'un programme linéaire est écrit sous une des formes suivantes :

1.3.1 Forme normale d'un programme linéaire

Tout programme linéaire peut s'écrire sous forme normale.Invalid source specified. [2]

$$\text{Max} \quad \sum_{i=1}^{n} c_i x_i$$

$$\text{Sous les contraintes} \quad \sum_{i=1}^{n} a_{ij} x_i \leq b_j \,, (j = 1, \dots, m)$$

$$x_i \geq 0, x_i \in \mathbb{R} \,, (i = 1, \dots, n)$$

Si on a une variable $x_i \in \mathbb{R}$, on introduit $x_i^+ \geq 0$ et $x_i^- \geq 0$ et on pose $x_i = x_i^+ + x_i^-$.

1.3.2 Forme standard d'un programme linéaire

Définition 1.3

Un programme linéaire est sous *forme standard* lorsque toutes ses contraintes sont des *égalités* et toutes ses variables sont non-négatives. Invalid source specified. [2]

$$\begin{cases} Max & \sum_{j=1}^{n} c_j x_j \\ \text{Sous} & \sum_{j=1}^{n} a_{ij} x_j = b_i \\ & x_j \geq 0 \end{cases}$$

- Représentation matricielle

$$\begin{cases} \text{Minimiser} & z = c^t x \\ \text{sujet à} & Ax = b \\ & x \geq 0 \\ \\ \quad \text{où} & c = [c_1 , c_2 , \dots , c_n]^t \;, \\ & b = [b_1 , b_2 , \dots , b_m]^t \\ \\ \quad et & A = [a_{ij} \,,\; i = 1, \dots , m \;\; ET\; j = 1, \dots , n] \end{cases}$$

Est une matrice de dimension m x n.

1.3.3 Forme canonique d'un programme linéaire

Définition 1.4

Un programme linéaire est *sous forme canonique* lorsque toutes ses contraintes sont des *inégalités* et toutes ses variables sont non-négatives. Invalid source specified. [2]

$$\begin{cases} Max & \sum_{j=1}^{n} c_j x_j \\ \text{Sous} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i \\ & x_j \geq 0 \end{cases}$$

- *Représentation matricielle*

$$\begin{cases} \textbf{Max} \quad z = c^t x \\ \\ \quad Ax \leq b \\ \quad x \geq 0 \end{cases}$$

Théorème 1 Tout programme linéaire peut s’écrire sous *forme standard* et sous *forme canonique.*

1.4 Résolution de programmes linéaires

Il existe plusieurs façons de résoudre les problèmes de programmation linéaire, Parmi eux :

- *Résolution graphique*
- *La méthode du simplexe*

Dans ce qui suit, nous allons décrire la méthode de simplexe.

2. La méthode de simplexe

Définition 2.1 : La méthode de simplexe est la plus célèbre méthode pour la résolution numérique des programmes linéaires. Analytiquement parlant, l’algorithme consiste à passer itérativement d’une solution de base réalisable non dégénérée x, à une solution de base réalisable non dégénérée voisine x’, en laquelle la valeur de Z est plus élevée, partant d’un point x_0. [3]

2.1 Les phases de la méthode du simplexe

La méthode du simplexe se décompose en deux phases essentielles :

Phase I : ***procédure d'initialisation***

Déterminer une première solution de base réalisable, i.e. les coordonnées d'un premier sommet de l'ensemble des solutions réalisables D. Si cette procédure échoue, cela signifie que D est vide, donc le problème est impossible. Sinon, le domaine des solutions réalisables n'est pas vide ; et soit x_0 une (première) solution de base réalisable (trouvée).

Phase II : ***procédure itérative***

Calculer, à partir d'une solution de base réalisable (trouvée en phase **I**), une autre solution de base réalisable donnant une valeur meilleure de la fonction objectif. Géométriquement, une itération consiste à passer d'un sommet de D à un autre sommet de D adjacent au précédent (dans le sens où elles sont les deux extrémités d'une arête de D). Leurs coordonnées correspondent à deux solutions de base réalisables dont les ensembles d'indices de base (donc aussi hors base) ne diffèrent que d'un seul indice. Cette caractéristique permettra de déterminer relativement aisément la nouvelle solution de base réalisable à partir de l'ancienne. [3]

2.2 L'application de l'algorithme simplexe

Pour pouvoir résoudre un tel programme avec la méthode de simplexe il fout d'abord que le programme soit t'écrié sous forme standard, comme indiqué dans la figure 1.

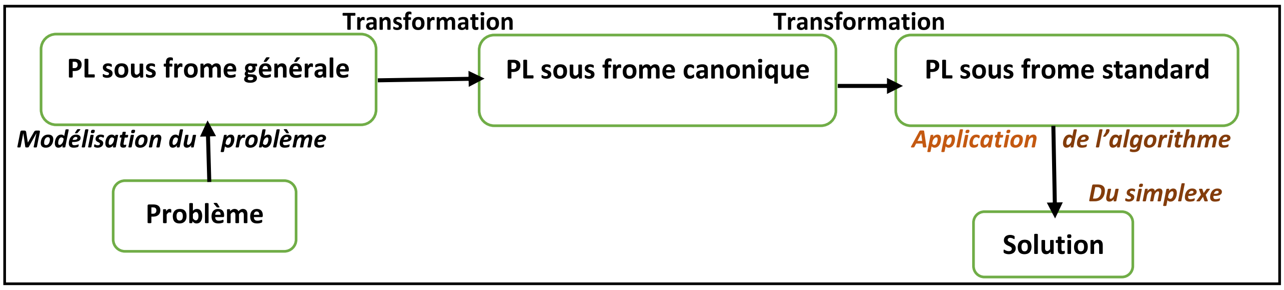

Figure 1. 1 Diagramme d'application d'algorithme simplexe

2.3 Les étapes de l'algorithme du simplexe

2.3.1 Algorithme simplex (cas de maximisation)

Étape 1 : *Formulation du modèle mathématique*

i. Formuler le modèle LP du problème donné.

ii. Si la fonction objectif est de minimisation, alors convertissez-la en maximisation équivalente, en utilisant la relation suivante Minimiser

$$Z = -\text{ Maximiser } Z^*, \text{ où } Z^* = -Z.$$

iii. Vérifier si toutes les valeurs b_i ($i = 1, 2, \ldots, m$) sont positives. Si l'un d'entre eux est négatif, alors multiplier la contrainte correspondante par -1 pour obtenir $b_i > 0$. Ce faisant, n'oubliez pas de changer une contrainte de type ≤ en une contrainte de type ≥, et vice versa.

iv. Exprimer le problème LP donné sous la forme standard en ajoutant des variables supplémentaires dans les contraintes (selon les besoins) et attribuez un coefficient de coût nul à ces variables dans la fonction objectif.

v. Remplacez chaque variable non restreinte (le cas échéant) par la différence des deux variables non négatives.

Étape 2 : *Configurer la solution initiale*

Notez les coefficients de toutes les variables du problème LP sous forme de tableau, comme indiqué dans le tableau 1, afin d'obtenir une solution réalisable initiale de base [$x_B = B^{-1}b$].

		$c_j \rightarrow$	c_1	c_2	$\ldots$	c_n	0	0	$\ldots$	0
Variables de base Coefficient (c_B)	Variables de base B	Variables de base valeur b (=x_B)	Variables x_1	x_2	$\ldots$	x_n	s_1	s_2		s_m
c_{B1}	S_1	$x_{B1} = b_1$	a_{11}	a_{12}		a_{1n}	1	0	…	0
c_{B2}	S_2	$x_{B2} = b_2$	a_{21}	a_{22}		a_{2n}	0	1	…	0
.	.	.	.	.		.	.	.		.
.	.	.	.	.		.	.	.		.
.	.	.	.	.		.	.	.		.
c_{Bm}	S_m	$x_{Bm} = b_m$	a_{m1}	a_{m2}		a_{mn}	0	0		1
Z= $\sum c_{Bi}x_{Bi}$		z_j=$\sum c_{Bi}x_j$	0	0		0	0	0	…	0
		c_j-z_j	$c_1 - z_1$	$c_2 - z_2$		$c_n - z_n$	0	0	…	0

Figure 1. 2 table simplex initiale

Après avoir mis en place la table simplex initiale, repérez la matrice identité. La matrice identité ainsi obtenue est aussi appelée matrice de base.

Les colonnes de la matrice d'identité représentent les coefficients des variables d'écart qui ont été ajoutés à les contraintes. Chaque colonne de la matrice d'identité représente également une variable de base.

Attribuez les valeurs des constantes ($b_i's$) aux variables de colonne dans la matrice d'identité [parce que x_B = B^{-1} b = I b = b].

Les variables correspondant aux colonnes de la matrice identité sont appelées variables de base et les autres sont appelées variables non fondamentales. En général, si un modèle *LP* a *n* variables et *m (< n)* contraintes, alors m variables seraient basiques et *n – m* variables non basiques. Dans certains cas, un ou plusieurs variables de base peuvent également avoir des valeurs nulles. Si une ou plusieurs variables de base ont une valeur nulle, alors cette situation est appelée dégénérescence.

La première ligne du tableau 1 indique les coefficients c_j des variables de la fonction objectif. Thèse les valeurs représentent le coût (ou profit) par unité associé à une variable dans la fonction objectif et ces servent à déterminer la variable à saisir dans la matrice de base B.

La colonne « c_B » répertorie les coefficients des variables de base actuelles dans la fonction objectif. Celles-ci les valeurs sont utilisées pour calculer la valeur de Z lorsqu'une unité de n'importe quelle variable est introduite dans la solution. La colonne 'x_B' représente les valeurs des variables de base dans la solution de base actuelle. Les nombres, a_{ij} dans les colonnes sous chaque variable sont également appelés taux de substitution (ou taux de change coefficients) car ceux-ci représentent le taux auquel la ressource i (i = 1, 2, . . ., m) est consommée par chaque unité d'une activité j (j = 1, 2, . . ., n).

Les valeurs z_j représentent le montant par lequel la valeur de la fonction objectif Z serait diminuée (ou augmenté) si une unité de la variable donnée est ajoutée à la nouvelle solution. Chacune des valeurs du $c_j - z_j$ représente le montant net d'augmentation (ou de diminution) de la fonction objective qui se produirait lorsqu'une unité de la variable représentée par l'en-tête de colonne est introduite dans la solution. C'est-à-dire :

$c_j - z_j$ (effet net) = c_j (bénéfice/coût unitaire entrant) – z_j (bénéfice/coût total sortant) où

z_j = Coefficient de la colonne des variables de base × Coefficient d'échange colonne j

Étape 3 : *test d'optimalité*

Calculez la valeur $c_j - z_j$ pour toutes les variables non fondamentales. Pour obtenir la valeur de $z_j$multiplier chaque élément sous la colonne « Variables » (colonnes, a_j de la matrice des coefficients) avec les éléments correspondants dans la colonne c_B. Examinez les valeurs de $c_j - z_j$.Les trois cas suivants peuvent se présenter :

i. Si tous $c_j - z_j \leq 0$, alors la solution réalisable de base est optimale.
ii. Si au moins une colonne de la matrice des coefficients (c.à.d. a_k) pour laquelle $c_k - z_k$>0 et tous les autres éléments sont négatifs (c.-à-d. $a_k < 0$), alors il existe une solution illimitée au problème donné.
iii. Si au moins un $c_j - z_j > 0$, et chacune de ces colonnes a au moins un élément positif (c.-à-d. a_{ij}) pour une ligne, cela indique qu'une amélioration de la valeur de la fonction objectif Z est possible.

Étape 4 : *Sélectionnez la variable pour entrer la base* Si le cas (iii) de l'étape 3 est vérifié, sélectionnez une variable qui a la plus grande valeur $c_j - z_j$ à entrer dans la nouvelle solution. C'est-à-dire,

$$c_k - z_k = \text{Max}\ \{(c_j - z_j)\ ;\ c_j - z_j > 0\}$$

La colonne à renseigner est appelée colonne clé ou colonne pivot. Évidemment, une telle variable indique la plus grande amélioration par unité dans la solution actuelle.

Étape 5 : *Test de faisabilité (variable à partir de la base)* Après avoir identifié la variable à devenir la variable de base, la variable à supprimer de l'ensemble existant de variables de base est déterminée. Pour cela, chaque nombre dans la colonne x_B (c.-à-d. les valeurs b_i) est divisé par le nombre correspondant (mais positif) dans la colonne clé et une ligne est sélectionnée pour laquelle ce rapport est non négatif et minimum. Ce rapport est appelé le taux de remplacement (échange). C'est-à-dire,

$$\frac{x_{Br}}{a_{rj}} = Min \left\{\frac{x_{Bi}}{a_{rj}} \; ; \; a_{rj} > 0 \right\}$$

Ce ratio limite le nombre d'unités de la variable entrante qui peuvent être obtenues à partir de l'échange. On peut noter que la division par un élément *négatif ou par un élément nul* dans la colonne clé n'est pas autorisée. La ligne sélectionnée de cette manière est appelée ligne clé ou *pivot* et elle représente la variable qui laissera la solution. L'élément qui se trouve à l'intersection de la ligne clé et de la colonne clé du tableau simplex est appelé clé ou *élément pivot*.

Étape 6 : *Trouver la nouvelle solution*

i. Si l'élément clé est 1, alors la ligne reste la même dans le nouveau tableau simplex.
ii. Si l'élément clé est différent de 1, alors divisez chaque élément de la ligne clé (y compris les éléments dans x_B-colonne) par l'élément clé, pour trouver les nouvelles valeurs pour cette ligne.
iii. Les nouvelles valeurs des éléments dans les lignes restantes du nouveau tableau simplex peuvent être obtenues en effectuant des opérations élémentaires sur toutes les lignes afin que tous les éléments sauf l'élément clé dans la colonne clé sont zéro. En d'autres termes, pour chaque ligne autre que la ligne clé.

$$E_{ij} = E_{ij} - \left(\frac{E_{ic}}{E_{Ic}}\right) * E_{Ij}$$ Où : *i : ligne* et *j : colonne*

I : *est l'indice de la ligne du pivot* et **c :** *l'indice de la colonne du pivot*

Étape 7 : *répétez la procédure* Passez à l'étape 3 et répétez la procédure jusqu'à ce que toutes les entrées du $c_j - z_j$ row sont soit négatifs soit nuls.Invalid source specified. [4]

Les étapes ci-dessus nous permet l'établissement de l'algorithme suivant :

Algorithme du simplexe

1) Initialement :

Le PL (P) $\begin{cases} Ax = b \\ cx = \max(z) - a^* \end{cases}$ $x \geq 0$ ***est écrit SFC% à une base réalisable J.***

Application *col :* {1, …, m}→ **J donnée. Matrice de coefficients : M**

$$\underbrace{x_{col(i)} = b_i(\forall_i = \overline{1,m}) \quad et \quad x_{\bar{J}} = 0}_{\textit{Solution de base réalisable associée à J (initiale).}} \qquad \underbrace{z(x) = a^*}_{\textit{Évaluation initiale}}$$

2) Choisir s tel que $c^s > 0$:

2.1) Si s n'existe pas. Terminer. J base optimale. Poser $x^* = x$

x^* Solution de base optimale associée à J ; $z(x) = a^*$ **évaluation optimale.**

2.2) sinon, aller à 3)

3) Soit $I = \{i | A_i^s > 0\}$:

3.1) Si $I = \emptyset$. *Terminer*. ***P () n'est pas de solution optimale.***

3.2) Sinon, aller à 4)

4) Soit $L = \left\{ l \middle| \frac{b_l}{A_l^s} = \operatorname{Min}_{i \in I} \left\{ \frac{b_i}{A_i^s} \right\} \right\}$; **Choisir** $r \in L$

Effectuer $Pivotage\ (m+1, n+1, r, s; M) = M'; J' = (J \cup \{s\}) \setminus \{col(r)\}$ **nouvelle base réalisable**

$$x'_s = \frac{b_r}{A_r^s};\ x'_{col(i)} = b_i - A_i^s x'_s\ ;\ x'_{\bar{J}} = 0\ ; a := a^* + c^s \frac{b_r}{A_r^s}$$ **nouvelle évaluation**

Poser $J := J', col(r) := s\ ; x := x';\ a^* := a\ ; M := M'$.aller à 2)

Fin de l'algorithme

Figure 1. 3 Algorithme du simplexe

Remarque : Pour le programme linéaire à fonction objectif à minimiser, on remplace dans l'algorithme l'instruction "Choisir s tel que $c^s > 0$" par "Choisir s tel que $c^s < 0$".

2.3.2 Organigramme de L'algorithme simplexe

Peut être résumée La méthode du simplexe, à la fois pour le problème de minimisation et de maximisation LP, par un organigramme comme indiqué dans la figure 3. Invalid source specified. [3]

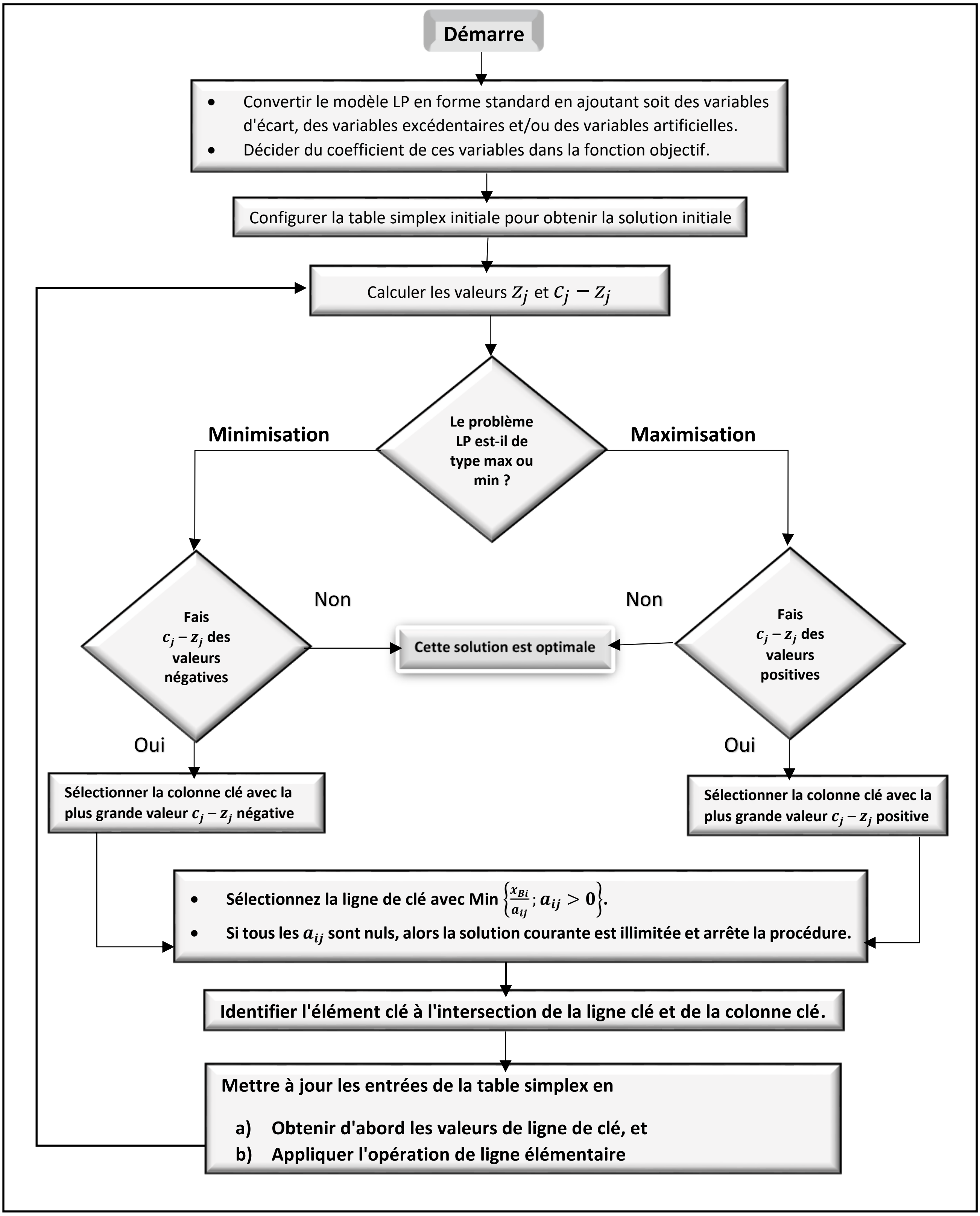

Figure 1. 4 Organigramme de L'algorithme simplexe

2.4 La dualité

Définition 2.2: La dualité dans la programmation linéaire stipule que chaque problème de programmation linéaire a un autre problème de programmation linéaire qui lui est lié et peut donc en être dérivé. Le problème de programmation linéaire d'origine est appelé "Primal", tandis que le problème linéaire dérivé est appelé "Dual". [3]

2.4.1 Dual d'un PL sous forme standard

Un programme linéaire est caractérisé par le tableau simplexe $\begin{bmatrix} A & b \\ c & \end{bmatrix}$

Par définition, le problème dual est obtenu en transposant ce tableau. $\begin{bmatrix} A^T & b^T \\ c^T & \end{bmatrix}$

Soit $v \in \mathbb{R}^n$ le vecteur-colonne des variables du problème dual ou $u \in \mathbb{R}^n$ le vecteur-ligne des variables du problème dual.Invalid source specified.

Le problème **primal** s'écrit :

$$(\text{P})\begin{cases} Min \ \ x^z & = cx \\ sous \ \ Ax & = b \\ \quad et \ \ x & \geq 0 \end{cases}$$

Le problème **dual** s'écrit :

$$(\text{D})\begin{cases} Max \ \ v \ \ W & = b^T v \\ sous \ \ A^T v & \leq c^T \\ \quad et \ \ v & \geq 0 \ ou \ < 0 \end{cases}$$

Ou encore, avec $u = v^T$,

$$(\text{D})\begin{cases} Max \ \ u \ \ W & = ub \\ sous \ \ uA & \leq c \\ \quad et \ \ u & \geq 0 \ ou \ < 0 \end{cases}$$

2.4.2 Dualité sous toutes ses formes

Dans le tableau suivant, on donne toutes les situations possibles.

	PL primal	PL dual
Variables	$x_1, x_2, \dots, x_n$	$y_1, y_2, \dots, y_m$
Matrice	A	A^T
Membre de droit	b	c
Fonction objectif	$Min\ c^T x$	$Max\ b^T y$
	$i$ème contrainte a $\leq$	$y_i \leq 0$
Contraintes	$\geq$	$y_i \geq 0$
	$=$	$y_i \in \mathbb{R}$
	$x_j \leq 0$	$j$ème contrainte a $\geq$
	$x_j \geq 0$	$\leq$
	$x_j \in \mathbb{R}$	$=$

Figure 1. 5 propriété importante de la dualité en PL

Exemple 2.1

Primal	Primal sous forme d'équation	Variables Dual
Maximiser $z = 5x_1 + 12x_2 + 4x_3$ Sujet à $x_1 + 2x_2 + x_3 \leq 10$ $2x_1 - x_2 + 3x_3 = 8$ $x_1, x_2, x_3 \geq 0$	Maximiser $z = 5x_1 + 12x_2 + 4x_3 + 0x_4$ Sujet à $x_1 + 2x_2 + x_3 + x_4 = 10$ $2x_1 - x_2 + 3x_3 + 0x_4 = 8$ $x_1, x_2, x_3, x_4 \geq 0$	 y_1 y_2

Problème **dual :**

Minimiser $w = 10y_1 + 8y_2$

Sujet à $y_1 + 2y_2 \geq 5$

$2y_1 - y_2 \geq 12$

$y_1 + 3y_2 \geq 4$

$$\left.\begin{array}{r} y_1 + 30 \geq 0 \\ y_1, y_2 \; sans\ restriction \end{array}\right\} \Longrightarrow (y_1 \geq 0, y_2 \; sans\ restriction).$$

3. Rappel sur les graphes

3.1 Qu'est-ce qu'un graphe

Un graphe G est un objet mathématique composé d'un ensemble V non vide et fini de points $\{v_1$, v_2, …$v_n\}$ appelés sommets (nœuds), et par un ensemble des couples de sommets noté E (qui peut être vide) de segments (flèches) e_1, e_2, …, e_m appelés arêtes (arcs), en reliant chaque paire de sommets v_1 et v_2 par une arête (arc) e, les points v_1 et v_2 appelés les extrémités.**Invalid source specified.**

On appelle ordre d'un graphe G le nombre de ses sommets, noté par |V (G)| Soit $x, y \in V$ on dit que x est adjacent à y si x et y sont reliés par une arête (arc). Si e = {x, y} est une arête, et si x = y alors e définit une boucle. En d'autres termes une boucle est une arête reliant un sommet à lui-même. **Invalid source specified.**

Le graphe H = (V, E') est un graphe partiel de G, si E' ⊆ E, Autrement dit on obtient H en enlevant une ou plusieurs arêtes au graphe G. Un sous graphe engendré par A ⊆ V est un graphe dont les sommets sont ceux de A et les arêtes sont celles ayant les deux extrémités dans A. on le note GA. [5]

Définition 3.1 : Un graphe G est le couple (X, E) talque :

X : ensemble des sommets de G.

E ⊆ (X * X) : ensemble des arcs de G

X = $\{x_1$, x_2, …$x_n\}$. $u \in E$, u=(x , y) , x , y ∈ X

x: extrémité initiale de u

y: extrémité terminale de u

Définition 2 : Un graphe dans lequel chacune des arêtes reliant deux sommets est appelé orientée (a un sens). Les arêtes d'un graphe orienté sont appelées des arcs.

Exemple : X = {A, B, C, D}, E = {20, 15, 13 ,10}.

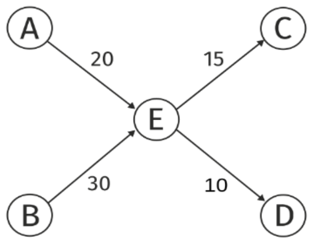

Figure 1. 6 graphe orienté

Définition 3 : Un graphe est dit biparti si son ensemble de sommets peut être divisé en deux sous-ensembles disjoints U et V tels que chaque arête ait une extrémité dans U et l'autre dans V.

Définition 4 : Un graphe est dit simple si deux sommets distincts sont joints par au plus une arête et s'il est sans boucle.

Définition 5 : Soit G= (X, U) un graphe orienté. On associe à chaque arc u= (x, y) une longueur $d(u)$: $d: u \to IR$ le graphe G muni de l'application d est appelé un réseau. Ce réseau est noté R= (X, U, d).

Conclusion

Dans ce chapitre, nous avons d'abord, rappelé quelques notions sur la programmation linéaire afin de poser les bases du problème de transport. Ensuite, nous avons présenté la méthode de simplexe.

La programmation linéaire est une méthode de la recherche opérationnelle. La programmation linéaire est un outil mathématique permettant de résoudre un modèle mathématique déterministe satisfaisant aux hypothèses de linéarité, d'additivité et de non-négativité des variables.

La méthode du simplexe est une des méthodes pour résoudre un programme linéaire.

Dans le chapitre suivant, nous appliquerons ce que nous avons abordé dans ce chapitre pour formuler le problème de transport.

Chapitre 2

Problème de Transport

Chapitre 2

Problème de Transport

Introduction :

C'est en 1941 que Frank L. Hitchcock a formulé pour la première fois le problème de transport, qui consiste à minimiser le coût de transport total d'un plan d'expédition. Le problème de transport « classique » est en fait un cas particulier d'un problème de flux de réseaux (minimiser à la fois la distance totale et le coût de transport).

Dans ce chapitre nous allons présenter et modéliser le problème de transport par ses différentes formulations.

1. Positionnement du problème :

On peut décrire un problème de transport de façon suivante : Une quantité donnée d'un produit uniforme est disponible à chacune des origines (par exemple des dépôts). Il s'agit d'en envoyer des quantités spécifiées à chacune des destinations (par exemple des points de vent). On connait le coût de transport d'une unité de l'une des origines vers l'une des destinations.

En supposant qu'il est possible d'expédier des produits depuis n'importe quelle origine vers n'importe quelle destination, il s'agit de déterminer le coût de transport minimum des origines vers les points de destination. [6]

Exemple

Société XYZ. Possède deux usines à différents endroits du pays où elles produisent des widgets. Leur partenaire commercial dispose de trois entrepôts centraux où ils expédient ces widgets à leurs différents clients. Les usines peuvent produire chacune un nombre donné de widgets par semaine et la demande prévue pour chaque entrepôt est également connue. Il y a un coût d'expédition de chaque usine à chaque entrepôt. Quelle usine devrait produire et expédier combien de widgets vers quels entrepôts pour répondre à la demande à chaque emplacement avec un coût minimal ? Cet exemple est considéré comme un "problème de transport"

2. Modélisation

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 donné. On nous donne également le coût de transport entre chaque paire d'entrepôt et de destination, telles que : [6]

– La disponibilité de chaque entrepôt i est : $a_i$unité, ou i = 1, 2, 3..., m.

– La demande de chaque destination j est : $b_j$unité, ou j = 1, 2, 3..., n.

– Le coût de transport d'une unité du produit de l'entrepôt i à la destination j est égal à c_{ij} unité, où i ∈{1, 2, 3..., m} et j ∈ {1, 2, 3..., n}.

Le coût total d'une expédition est linéaire en taille d'expédition.

Variables de décision

Les variables du modèle de programmation linéaire (PL) 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 la source i vers la destination j, où i ∈{1, 2, 3..., m} et j ∈ {1, 2, 3..., n}.

Fonction objective

Le problème consiste à déterminer les quantités x_{ij} à transporter de façon que le coût total de transport $\sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij}x_{ij}$ soit minimal.

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

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 sommet. Posons : a_i désigne une capacité d'une source (disponibilité) et b_j désigne le besoin d'une destination (demande)

Les contraintes sont :

– La disponibilité à chaque source doit être épuisée : $\sum_{j}^{n} x_{ij} = a_i$, i ∈{1, 2, 3..., m}.
– La demande à chaque destination doit être satisfaite : $\sum_{i}^{m} x_{ij} = b_j$,j ∈ {1, 2, 3..., n}.
– La non négativité des quantités : $x_{ij} \geq 0, \forall_i \in \{1, ..., m\}$; $\forall_j \in \{1, ...,n\}$

3. Formulation du problème de transport :

Un problème de transport peut être formulation de trois manières :

– Sous la forme d'un tableau de transport

– Sous la forme d'un programme linéaire

– Sous la forme d'un graphe biparti

3.1 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, de la demande et le coût de transport entre chaque source et destination. [7]

La figure 2.1 présente un tableau de problème de transport :

	destinations				
Origines	D_1	D_2	.. ..	D_n	Offre
$Origine_1$	$C_{11}\ X_{11}$	$C_{12}\ X_{12}$	.. ..	$C_{1n}\ X_{1n}$	$offre_1$
$Origine_2$	$C_{21}\ X_{21}$	$C_{22}\ X_{22}$	.. ..	$C_{2n}\ X_{2n}$	$offre_2$
.. ..	.. ..	.. ..	.. ..	.. ..	.. ..
$Origine_m$	$C_{m1}\ X_{m1}$	$C_{m2}\ X_{m2}$	.. ..	$C_{mn}\ X_{mn}$	$offre_m$
Demande	$demande_1$	$demande_2$	.. ..	$demande_n$	$\sum d_i = \sum o_j$

Figure 2. 1 Tableau de transport

3.2 Par un programme linéaire :

Min Z =$\sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij}x_{ij}$

$$SC \begin{cases} \sum_{i=1}^{m} a_{ij} = \sum_{j=1}^{n} b_j \ \forall_i = 1,,m \ et \ \forall_j = 1,,n \\ \sum_{j=1}^{n} a_i \ \ i = 1,,n \\ \sum_{i=1}^{m} b_j \ \ j = 1,,m \\ a_i \in \mathrm{N} \ \forall_i = 1,,m \ ; \ b_j \ \in \mathrm{N} \ \forall_j = 1,,n \\ et \ X_{ij} \in \mathrm{N} \ \forall_i = 1,,m \ et \ \ \forall_j = 1,,n \end{cases}$$

Fonction :

Le problème consiste à déterminer les quantités x_{ij} à transporter de façon que le coût total de transport $\sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij}x_{ij}$ soit minimal.

Variables :

$\boldsymbol{x_{ij}}$: La quantité à transporter de $\boldsymbol{i}$ vers $\boldsymbol{j}$ $\forall_{ij} \in \{\mathbf{1}, \ldots, \boldsymbol{n}\} \times \{\mathbf{1}, \ldots, \boldsymbol{m}\}$.

$\boldsymbol{c_{ij}}$: Le coût unitaire de transporter de $\boldsymbol{i}$ vers $\boldsymbol{j}$ $\forall_{ij} \in \{\mathbf{1}, \ldots, \boldsymbol{n}\} \times \{\mathbf{1}, \ldots, \boldsymbol{m}\}$.

Contraintes :

- ✓ La quantité totale de marchandises partant de $\boldsymbol{i}$ est égale à la disponibilité $\sum_{j=1}^{n} x_{ij} = a_i$; $\forall_i$=1,,n.
- ✓ La quantité totale de marchandises reçue par $\boldsymbol{j}$ est égale à la demande $\sum_{i=1}^{m} x_{ij} = b_j$; $\forall_j$=1,,m.

Remarque :

- ➢ Pour résoudre le problème, on doit toujours vérifier que le total des quantités disponibles correspond au total des quantités demandées, c'est-à-dire que l'offre globale soit égale à la demande globale :
  $\sum_{i=1}^{m} a_{ij} = \sum_{j=1}^{n} b_{ji}$ $\forall_i$=1 ,,m et $\forall_j$=1,,n
  On dit que le problème **est équilibré.**
- ➢ Si ce n'est pas le cas, par exemple si :
  - ❖ $\sum_{i=1}^{m} a_{ij} > \sum_{j=1}^{n} b_{ji}$ $\forall_i$=1 ,,m et $\forall_j$=1,,n , l'offre est supérieure à la demande [excès des disponibilités], il faut créer une destination fictive avec un offre $\boldsymbol{a}$ et

attribuer un coût énorme ou nulle pour qu'aucune demande réelle ne puisse satisfaire cette destination fictive.

- $\sum_{i=1}^{m} a_{ij} > \sum_{j=1}^{n} b_{ji}$ $\forall_i$=1 ,,m et $\forall_j$=1,,n, la demande est supérieure à l'offre [excès des demandes], il faut créer une origine fictive avec une demande $\boldsymbol{b}$ et attribuer un coût énorme ou nulle pour qu'aucun offre réel ne puisse satisfaire cette origine fictive.

3.3 Graphe biparti

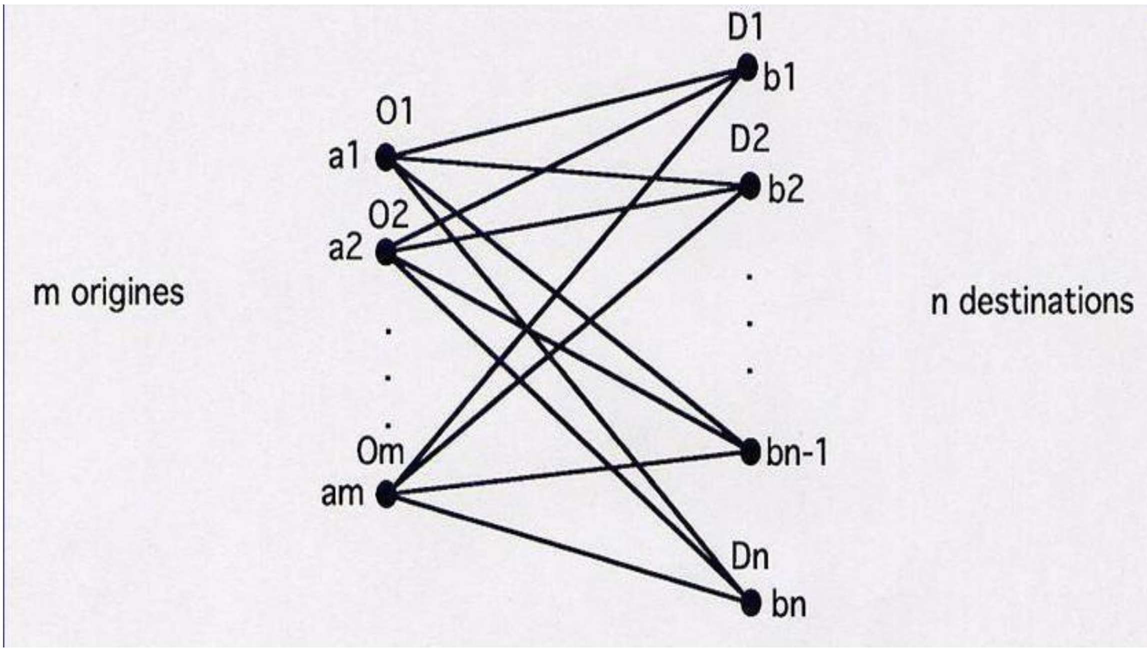

Figure 2. 2- Représentation du problème sous forme de graphe biparti

Chaque origine est représentée par un sommet O_i ,chaque destination par un sommet D_j , chaque route de l'origine i à la destination j par un arc orienté de $O_i\ vers\ D_j$.

4. 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. [7]

5. Exemple

Formulation sous la forme d'un tableau de transport :

Soit une série de villes alimentées en électricité par des centrales. La situation est résumée par la table suivante :

	A	B	C	D	Puissance fournie (GWh)
1	6	5	3	1	**500**
2	10	8	4	2	**300**
3	7	9	11	12	**200**
Demande (GWh)	**300**	**300**	**300**	**100**	

Figure 2. 3 exemple un tableau de transport

La structure d'un tableau de transport est assez intuitive comme le montre l'exemple de la Figure 2.3.
Dans ce problème, on a trois origines et quatre destinations. Les offres des origines sont inscrites sur la dernière colonne, et les quantités disponibles dans les différentes destinations sont inscrites sur la dernière ligne. Les chiffres inscrits en petite taille dans chaque case indiquent les coûts de transport unitaires entre chaque origine et chaque destination. Par exemple, chaque unité transportée de l'origine 2 vers la destination 3 induit un coût de transport de 4(um). Remarquons que dans ce tableau l'offre totale est égale à la demande totale. On dit que ce problème est *équilibré*. Si le problème n'est pas équilibré, on est dans le cadre d'un cas particulier qu'on discutera à la fin de ce cours.

Formulation sous la forme d'un programme linéaire

1) Définition des variables

$x_{ij} = nomber\ de\ GWh\ produits$ à $la\ centrale\ i\ et\ envoy$é à $la\ cité\ j$

2) Description de la fonction économique

Min z=

$$6_{x11} + 5_{x12} + 3_{x13} + 1_{x14}$$
$$+10_{x21} + 8_{x22} + 4_{x23} + 2_{x24}$$
$$+7_{x31} + 9_{x32} + 11_{x33} + 12_{x34}$$

3) Les contraintes

$$contraintes\ de\ production \begin{cases} x11 + x12 + x13 + x14 \leq 500 \\ x21 + x22 + x23 + x24 \leq 300 \\ x31 + x32 + x33 + x34 \leq 200 \end{cases}$$

$$contraintes\ de\ consommation \begin{cases} x11 + x21 + x3 \geq 300 \\ x12 + x22 + x32 \geq 300 \\ x13 + x23 + x33 \geq 300 \\ x14 + x24 + x34 \geq 100 \end{cases}$$

Plus les contraintes non négativité ($x_{ij} \ \geq 0$)

Formulation sous la forme de graphe bipartie

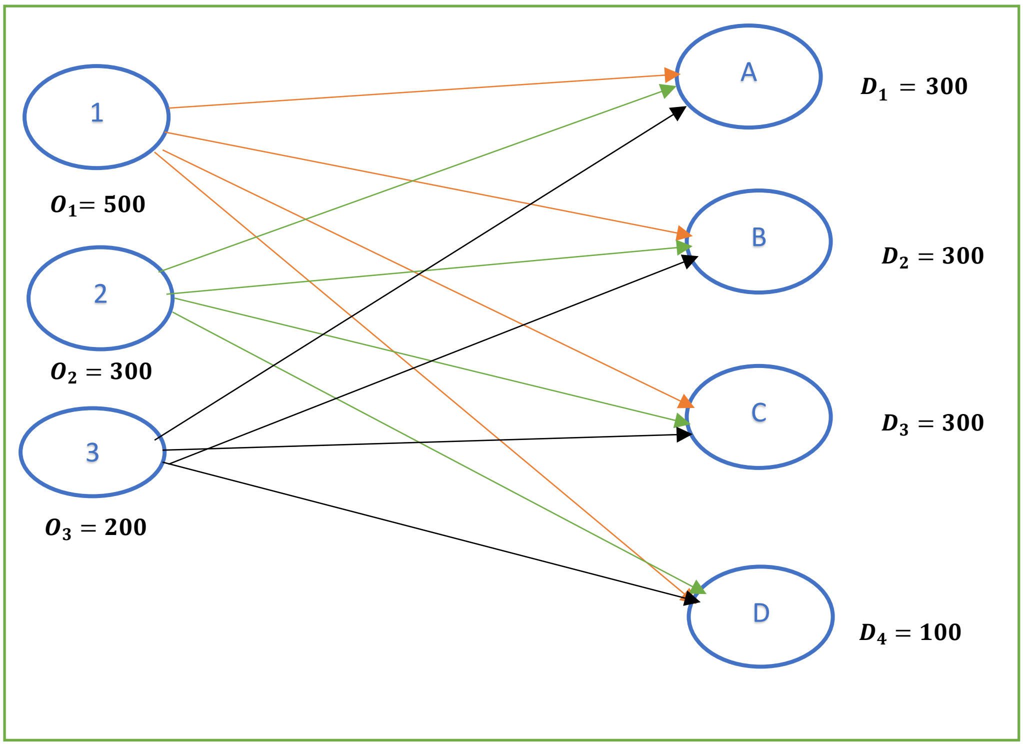

Figure 2. 4 Exemple formulation sous la forme de graphe bipartie

6. Le dual du problème de transport

Le dual du problème de transport est important parce que nous utilisons des variables duales dans calcul des coûts réduits dans le primaire problème.

$$max \sum_{i=1}^{m} S_i\, u_i + \sum_{j=1}^{n} D_j\, v_j$$

Sujet à :

$$u_i + v_j \leq c_{ij}, pour\ tout\ i, j$$

$$u_i \text{ et } v_j \text{ sans restriction.}$$

7. Conclusion

Le problème de transport est un type spécial de problème de programmation linéaire, où l'objectif est de minimiser le coût de distribution d'un produit à partir d'un certain nombre de sources vers un certain nombre de destinations.

Le transport concerne le transport d'une marchandise (produit unique) à partir de « m » sources (origine ou centres d'approvisionnement ou de capacité) vers « n » destinations (puits ou centres de demande ou de besoin).

On suppose que le niveau d'approvisionnement de chaque source et le montant de la demande à chaque destination sont connu. Le coût unitaire de transport des produits de chaque source à chaque destination est connu.

L'objectif est de déterminer la quantité à déplacer de chaque source vers chaque destination de sorte que le coût total du transport est minimum.

Dans le chapitre suivant, nous apporterons des solutions au problème de transport.

Chapitre 3

Résolution de Problème de transport

Chapitre3

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.

Dans ce chapitre nous abordons les méthodes de résolution de problème de transport. Nous allons présenter les méthodes : Coin Nord-Ouest, coût minimum, approximation de Vogel pour la recherche de la solution de base initiale, et les méthodes : Stepping-Stone, distribution modifiée pour la recherche de la solution optimale.

1. Structure de la résolution de problème de transport

La résolution du problème passe par deux étapes essentielles :

— La première c'est de trouver une solution de base initiale.

— La deuxième étape est de trouver la solution optimale à partir de la solution de base.

Pour résoudre le problème de transport, les étapes suivantes sont à suivre systématiquement :

1-Obtenir la solution faisable initiale, c'est-à-dire identifier la solution qui satisfait aux exigences de la demande et de l'offre. Il existe plusieurs méthodes par lesquelles la solution réalisable initiale peut être obtenue ce sont : -Méthode de Coin Nord-Ouest. -Méthode du Coût Minimum. -Méthode d'Approximation de Vogel.

2-Tester l'optimalité de la solution faisable initiale. Une fois la solution réalisable obtenue, l'étape suivante consiste à vérifier si elle est optimale ou non. Il existe deux méthodes utilisées pour tester l'optimalité :

-Méthode Stepping-Stone.

-Méthode de distribution modifiée. [8]

1.1 Solution de base réalisable

Définition 3.1 :

On appelle solution de base réalisable d'un programme de transport, une solution admissible comportant $M = (\mathrm{m} + \mathrm{n} - 1) \quad x_{ij} > 0$, c'est-à-dire qu'une solution de base comporte $(\mathrm{m}.\mathrm{n} - \mathrm{M})$zéros . Le graphe d'une solution de base est un graphe connexe sans cycle, c'est-à-dire un arbre comportant $\mathrm{N} = \mathrm{m} + \mathrm{n}$ sommets soit $M = \mathrm{N} - 1$ arcs. Si le nombre d'allocations dans les solutions de bases réalisables est inférieur à $(M + \mathrm{n} - 1)$, on appelle une solution de base dégénérée.

Interprétation en termes de théories de graphes de la notion de solution de base

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 $\varphi_{ij} = \min\{a', b'\} = (a'_i, b'_j \neq 0\ , \varphi_{ij} \in N)$ avec :

a' = disponibilité restante dans x_i

b' = 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 $(\mathrm{m} = \mathrm{n} - 1)$ flux φ_{ij} non nuls et nous obtenons une solution de base : $(\mathrm{m} \times \mathrm{n} - (\mathrm{m} + \mathrm{n} - 1) = (\mathrm{m} - 1)(\mathrm{n} - 1))$ flux nuls.

1.2 Solution optimale

Définition 3.2 :

Une solution réalisable est dite optimale si elle minimise le coût total de transport Problème de transport équilibré - un problème de transport dans lequel l'offre totale de toutes les sources est égale aux demandes totales dans toutes les destinations.

1.3 Organigramme de résolution pour le problème de transport

1) Tout d'abord, le problème est formulé comme un tableau 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.

On résume la résolution de problème de transport sous forme d'organigramme suivant (figure 3.1)

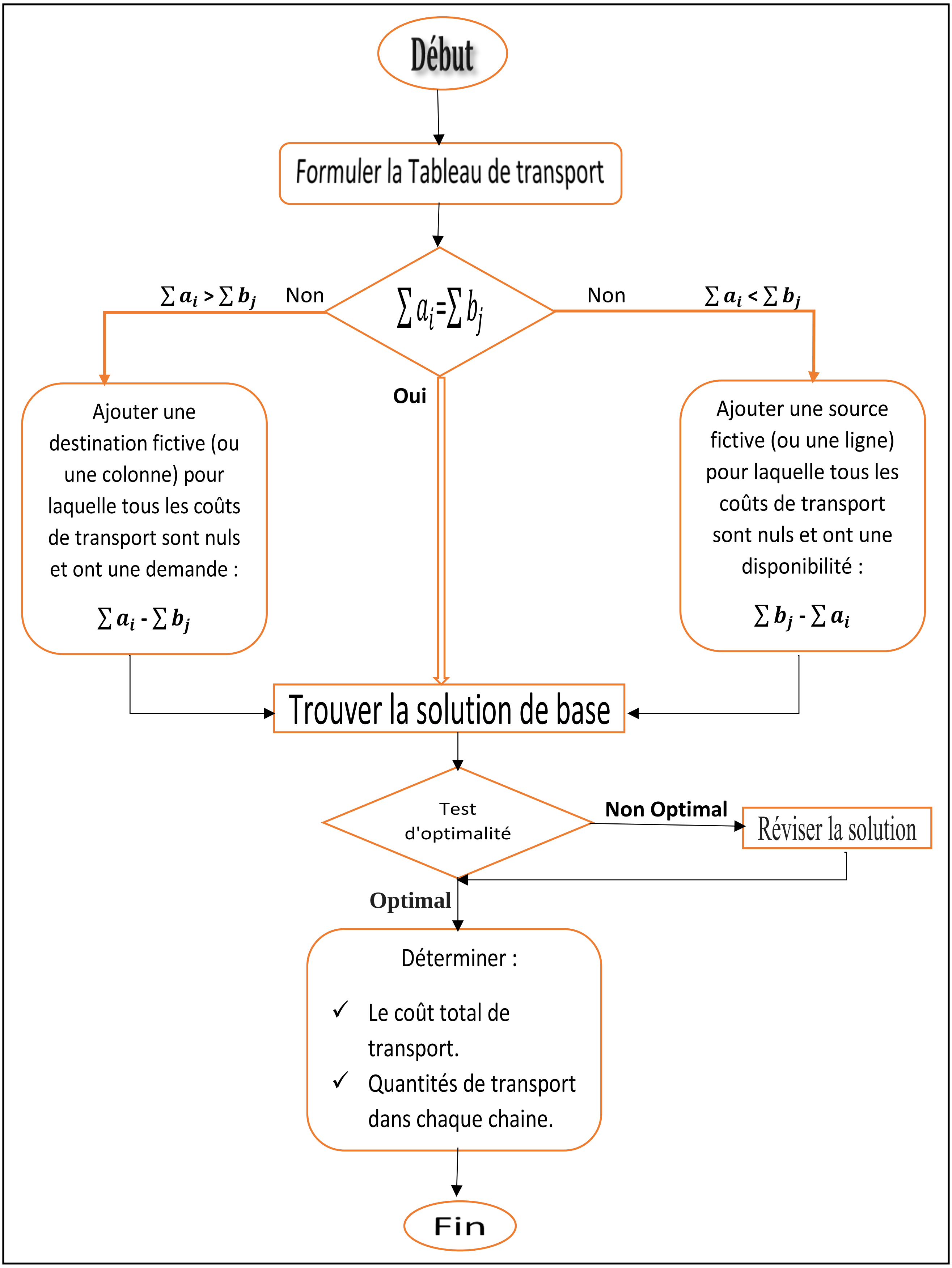

Figure 3. 1 Organigramme de la résolution de problème de transport.

1.4 Algorithme général de résolution de problème de transport

Les étapes de base de l'algorithme de transport sont exactement celles de la méthode du simplexe (Chapitre 1). Cependant, au lieu d'utiliser le tableau simplex régulier, nous profitons de la structure particulière du modèle de transport pour effectuer l'algorithme calculs plus facilement.

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

Étape 1 :

Construire la matrice de transport à partir du problème de transport donné : 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, et passez à l'étape 2.

Étape 2 :

Déterminez une solution réalisable de base de départ et passez à l'étape 2. 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.

– Méthode de simplexe adapté.

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. Recherche de la solution réalisable de base initiale

La procédure de calcul d'une solution réalisable de base initiale est effectuée dans un tableau de même dimension que le tableau des coûts de transport ; le transport tableau de solution, où chaque position (i, j) est associée à la décision variable x_{ij} , c'est-à-dire le nombre d'unités de produit à transporter depuis l'origine O_i vers la destination D_j . Ces positions (i, j) sont appelées cellules, et représenter une solution. Une cellule vide indique une valeur de zéro. Comme la figure 3.2.

	D_1	D_2	…	D_n	Offre
O_1	x_{11}	x_{12}	…	x_{1n}	a_1
O_2	x_{21}	x_{22}	…	x_{2n}	a_2
. . .	. . .	. . .	…	. . .	. . .
O_m	x_{m1}	x_{m2}	…	x_{mn}	a_m
Demande	b_1	b_2	…	b_n	

Figure 3. 2 Le tableau des solutions du problème de transport

Nous allons maintenant donner une explication des méthodes d'obtenir la solution de base de problème de transport.

2.1 Méthode du Coin Nord-Ouest

Définition 3.4 :

C’est une heuristique qui est appliquée à la programmation linéaire, c’est la méthode la plus rapide et la plus simple pour déterminer une solution de base car elle ne fait pas entrer les coûts de transport c’est à cette raisons là que généralement la solution obtenue par cette méthode est loin de la solution optimale. En optimisation combinatoire, une heuristique est un algorithme approché qui permet d’identifier en temps polynomial au moins une solution réalisable rapide, pas obligatoirement optimale. L’usage d’une heuristique est efficace pour calculer une solution approchée d’un problème et ainsi accélérer le processus de résolution exacte. [9]

Principe :

Cette méthode consiste à partir du coin supérieur gauche (le coin nord-ouest sur une carte géographique) du tableau en suivant les étapes suivantes : Allouer le plus possible à la cellule courante et ajuster l’offre et la demande ; Se déplacer d’une cellule vers la droite (demande nulle) ou le bas (offre nulle) ; Répéter jusqu’au moment où toute l’offre est allouée.

Algorithme de Coin Nord-Ouest :

Étape 1

Dans les lignes et les colonnes considérées, sélectionnez la cellule (i, j) dans le coin supérieur gauche (coin nord-ouest) du tableau des solutions (pour commencer, je = 1, j = 1).

Étape 2

Attribuez à la variable x_{ij}le montant maximal possible compatible avec les exigences de ligne et de colonne de cette cellule, c'est-à-dire la valeur :

$$x_{ij} = \min\{a_i, b_j\}.$$

Au moins un des besoins, l'offre ou la demande, sera alors rencontré. Ajustez l'offre a_i et la demande b_j comme suit :

• Si ai est le minimum, alors la fourniture de l'origine O_i devient zéro et la ligne i est éliminée de toute considération ultérieure. La demande b_j est remplacée par $b_j - a_i$.

• Si b_j se trouve être le minimum, alors la demande de la destination D_j devient zéro et la colonne j est éliminée de toute considération ultérieure. L'offre a_i est remplacée par $a_i - b_j$.

• Si $a_i = b_j$, alors les valeurs ajustées de l'offre a_i et de la demande b_j deviennent tous les deux nuls. La ligne i et la colonne j sont éliminées de plus de considération.

Étape 3

Deux cas peuvent se présenter :

• Si une seule ligne ou une seule colonne reste à l'étude, alors toutes les cellules restantes (i, j), c'est-à-dire les variables x_{ij} associées à ces cellules, sont sélectionnées et les fournitures restantes leur sont affectées. Arrêt.

• Sinon, passez à l'étape 1.

2.2 Méthode d'approximation de Vogel

Définition 3.5 :

La méthode d'approximation de Vogel's ou VAM est une procédure itérative calculée pour trouver la solution réalisable initiale du problème de transport. Comme la méthode du moindre coût, ici aussi les frais de port sont pris en considération, mais dans un sens relatif. [9]

Principe :

La principale différence entre la méthode du coin nord-ouest et l'approximation de Vogel méthode repose sur les critères utilisés à l'étape 1 pour sélectionner une cellule dans la solution tableau. Au lieu de sélectionner le coin supérieur gauche, l'approximation de Vogel La méthode calcule les différences de lignes et les différences de colonnes pour sélectionner une cellule. Le la différence de ligne et la différence de colonne sont définies comme suit :

• RD_i = la différence arithmétique entre le plus petit et le plus petit suivant coût unitaire c_{ij} qui restent à l'étude à la ligne i , $\mathrm{i} = 1, \ldots \mathrm{m}$.

• CD_j = la différence arithmétique entre le plus petit et le plus petit suivant coût unitaire c_{ij} qui restent à l'étude dans la colonne j, $\mathrm{j} = 1, \ldots, \mathrm{n}$.

Algorithme d'approximation de Vogel :

Les différences de ligne et les différences de colonne sont utilisées pour rendre plus pratique sélection d'une cellule dans le tableau des solutions ; une sélection basée sur l'unité les coûts de transport. Étant donné un problème de transport équilibré, et à partir d'un tableau de solutions avec toutes les cellules (i, j) vides, la méthode d'approximation de Vogel consiste des étapes suivantes, et conduit à une solution réalisable de base raisonnablement bonne :

Étape 1

Calculez pour chaque ligne i et chaque colonne j les différences RD_i et CD_j. Parmi les lignes et colonnes encore à l'étude, retrouvez celle avec la plus grande différence, et y trouver la cellule (i, j) avec la plus petite unité coût de transport c_{ij}.

Étape 2

Attribuez à la variable x_{ij} le montant maximal possible compatible avec les exigences de ligne et de colonne de cette cellule, c'est-à-dire la valeur :

$$x_{ij} = \min\{a_i, b_j\}.$$

Au moins un des besoins, l'offre ou la demande, sera alors rencontré. Ajustez l'offre a_i et la demande b_j comme suit :

• Si a_i est le minimum, alors la fourniture de l'origine O_i devient zéro et la ligne i est éliminée de toute considération ultérieure. La demande b_j est remplacée par $b_j - a_i$.

• Si b_j se trouve être le minimum, alors la demande de la destination D_j devient zéro et la colonne j est éliminée de toute considération ultérieure. L'offre ai est remplacée par $a_i - b_j$.

• Si $a_i = b_j$, alors les valeurs ajustées de l'offre a_i et de la demande b_j deviennent tous les deux nuls. La ligne i et la colonne j sont éliminées de plus de considération.

Etape 3

Deux cas peuvent se présenter :

• Si une seule ligne ou une seule colonne reste à l'étude, alors toutes les cellules restantes (i, j) , c'est-à-dire les variables x_{ij} associées à ces cellules, sont sélectionnées et les fournitures restantes leur sont affectées. Arrêt.

• Sinon, passez à l'étape 1.

2.3 Méthode du Coût Minimum

Définition 3.6 :

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. [9]

Principe :

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.

Algorithme du Coût Minimum :

Étape 1

Trouver la cellule (i, j), telle que c_{ij} est le plus petit coût de tout le tableau.

Étape 2

Envoyer le maximum d'unités pour la cellule (i, j). Ainsi x_{ij} est initialisé comme étant le $\min\{a_i, b_j\}$. Ajuster ensuite a_i etb_j, en tenant compte du montant x_{ij} à expédier. Exprimons cette phrase à l'aide d'inégalités :

$$x_{ij} = \min\{a_i, b_j\}.$$

$$a_i' = a_i - x_{ij}$$

$$b_j' = b_j - x_{ij}$$

Entourer (ou mettre en évidence d'une autre manière) le coût c_{ij}. À la fin de cette étape, soit a_i', soit b_j' est nul, soit les deux.

Étape 3

- Si $a_i' = 0$ et $b_j' > 0$, cela signifie que l'origine i a été "vidée". Il faut donc éliminer la ligne i du tableau.
- Si $b_j' = 0$ et $a_i' > 0$, cela signifie que la destination j est entièrement satisfaite et qu'il reste des marchandises dans le dépôt i. Il faut donc éliminer la colonne j du tableau.
- Si $a_i' = 0$ et $b_j' = 0$, nous nous trouvons dans un cas dégénéré. On élimine alors la ligne i, à moins qu'elle ne soit la seule ligne restante du tableau ; auquel cas il faut éliminer la colonne j.

Étape 4

- S'il reste un total de deux ou plusieurs lignes et colonnes non encore éliminées, reprendre à l'étape 1.
- S'il ne reste qu'une ligne non éliminée, la solution réalisable de base initiale est déterminée par les cellules entourées. [10]

3. Recherche de la solution réalisable de base optimale

Nous allons maintenant étudier des méthodes permettant de trouver la solution réalisable de base optimale.

Une fois que nous obtenons la solution réalisable de base pour un problème de transport, la tâche suivante consiste à tester si la solution obtenue est-elle optimale ou non ? Cela peut être fait par deux méthodes. (a) méthode Stepping-Stone, et (b) par méthode de distribution modifiée, ou méthode MODI.

3.1 Méthode de Stepping-Stone

Définition 3.7:

L'idée du Stepping-Stone est de partir d'une solution de base réalisable (non optimale) afin de l'améliorer itérativement jusqu'à obtenir une solution non optimisable. Il n'existe pas de meilleure solution, donc elle est optimale. [9]

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

Principe :

Le principe de cette méthode est de partir d'une solution de base et de progresser par itération pour trouver une solution qui minimise les coûts de transport.

- **Organigramme de la méthode Stepping-Stone**

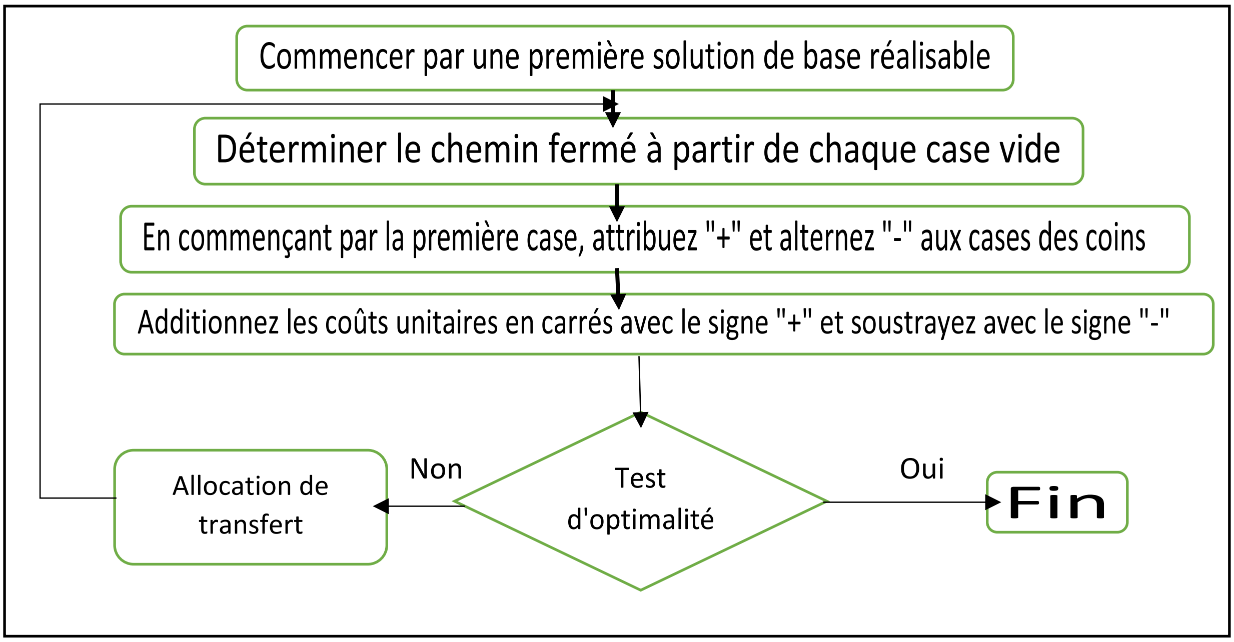

Figure 3. 3 Organigramme de la méthode Stepping-Stone

- **Algorithme de Stepping Stone :**

  - ***Déroulement de l'algorithme :***

L'algorithme de Stepping Stone consiste à modifier la solution de base vers une solution meilleure, donc à rendre non vide une case vide dans le tableau des quantités.

On appelle :

$T_i \ll u_i \gg$: potentiel origine.

$T_j \ll v_j \gg$: potentiel destination.

α_{ij}: coût marginal de la liaison (x_i, x_j).

  - ***Étapes de l'algorithme Stepping Stone :***

Étape 1 : Établir le tableau de transport initial.

Étape 2 : Déterminer une solution réalisable de base initiale.

Étape 3 : Assurez-vous que le nombre de cellules occupées est exactement égal à m+n-1, où m est le nombre de lignes et n est le nombre de colonnes.

Étape 4 : Soustrayez chaque minimum de ligne des coûts de ligne correspondants et chaque minimum de colonne des coûts de colonne correspondants.

Étape 5 : Calculer les valeurs de boucles associées aux cellules hors de la base.

(a) Si toutes ces valeurs sont non-négatives, la procédure est achevée, la solution réalisable de base courante est optimale.

(b) Si, pour une cellule hors de la base, on obtient une valeur de boucle négative, celle-ci est candidate pour entrer dans la base ; passer alors à la phase 6.

- La valeur d'une boucle, ou évaluation d'une cellule, est déterminée de la façon suivante :

Soit B(i, j) la boucle unique formée lorsque la cellule (i, j) est introduite dans la base. En partant de (i, j) , dans l'une ou l'autre des directions, nous parcourons la boucle en nommant alternativement les cellules soit recevantes, soit cédantes, le tout en considérant (i, j) , comme cellule recevante. La valeur de B(i, j) ,notée v(i, j) est obtenue en additionnant les coûts des cellules recevantes et en soustrayant les coûts des cellules cédantes.

Étape 6 : Introduire une cellule candidate (i, j) dans la base à l'aide de la méthode d'entrée. Reprendre ensuite à l'étape 5.

➢ ***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 3.8 :

Est une version modifiée de la méthode de stepping stone dans laquelle les équations mathématiques remplacent les chaînes de substitutions. Cette méthode est plus pratique que stepping stone. La méthode de distribution modifiée fournit une solution à coût minimum au problème de transport. Dans la méthode du tremplin, nous devons dessiner autant de chemins fermés que de cellules inoccupées pour leur évaluation. Au contraire, dans la méthode MODI, seul le chemin fermé pour la cellule inoccupée avec le coût d'opportunité le plus élevé est tracé. [9]

❖ **Étapes de la Méthode de Distribution Modifiée**

Étape 1 : Déterminer une solution réalisable de base initiale.

Étape 2 : Déterminer les valeurs des variables duales, u_i et v_j, en utilisant $c_{ij} = u_i + v_j$

Étape 3 : Calculez le coût d'opportunité en utilisant $\Delta_{ij}= c_{ij}-(u_i + v_j)$.

Étape 4 : Vérifiez le signe de chaque coût d'opportunité. Si les coûts d'opportunité de toutes les cellules inoccupées sont positifs ou nuls, la solution donnée est la solution optimale. D'autre part, si une ou plusieurs cellules inoccupées ont un coût d'opportunité négatif, la solution donnée n'est pas une solution optimale et des économies supplémentaires sur les coûts de transport sont possibles.

Étape 5 : Sélectionnez la cellule inoccupée avec le plus petit coût d'opportunité négatif comme cellule à inclure dans la solution suivante.

Étape 6 : Dessinez un chemin fermé ou une boucle pour la cellule inoccupée sélectionnée à l'étape précédente. Veuillez noter que le virage à angle droit dans ce chemin n'est autorisé qu'aux cellules occupées et à la cellule inoccupée d'origine.

Étape 7 : Attribuez des signes plus et moins alternatifs aux cellules inoccupées sur les points d'angle du chemin fermé avec un signe plus à la cellule en cours d'évaluation.

Étape 8 : Déterminez le nombre maximum d'unités qui doivent être expédiées vers cette cellule inoccupée. La plus petite valeur avec une position négative sur le chemin fermé indique le nombre d'unités pouvant être expédiées vers la cellule entrante. Maintenant, ajoutez cette quantité à toutes les cellules sur les points d'angle du chemin fermé marqués de signes plus et soustrayez-la de ces cellules marquées de signes moins. De cette manière, une cellule inoccupée devient une cellule occupée.

Étape 9 : Répétez toute la procédure jusqu'à ce qu'une solution optimale soit obtenue.

- ***Résumé des étapes de La méthode de distribution modifiée (MODI)***

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.

4. Exemple pratique

Soit une série de villes alimentées en électricité par des centrales. La situation est résumée par la table suivante :

	A	B	C	D	Puissance fournie (GWh)
1	6	5	3	1	**500**
2	10	8	4	2	**300**
3	7	9	11	12	**200**
Demande (GWh)	**300**	**300**	**300**	**100**	

4.1 Résolution du Problème

1) Solution initiale

a. La méthode du coin Nord-Ouest

Partir du coin supérieur gauche du tableau.

1. Allouer le plus possible à la cellule courante et ajuster l'offre et la demande ;

2. Se déplacer d'une cellule vers la droite (demande nulle) ou le bas (offre nulle) ;

3. Répéter jusqu'au moment où toute l'offre est allouée et toute la demande est satisfaite.

L'application à l'exemple proposé plus haut nous donne le tableau représenté dans la Figure 3.4.

	A	B	C	D	Offre
1	300 6	200 5	3	1	**500**
2	10	100 8	200 4	2	**300**
3	7	9	100 11	100 12	**200**
Demande	**300**	**300**	**300**	**100**	

Figure 3. 4 Solution initiale (coin Nord-Ouest)

- Le coût total correspondant à cette solution est de 6700(um).

b. La méthode du Coût Minimum

1. Repérer la case du tableau ayant le coût le plus faible ;
2. Affecter à cette case la quantité maximale possible ; une colonne ou une ligne est saturée ;
3. Si une colonne est saturée, l'éliminer du tableau, mettre à jour la quantité dans la ligne correspondante et reprendre au point 1 avec le nouveau tableau ;
4. Si une ligne est saturée, l'éliminer du tableau, mettre à jour la quantité dans la colonne correspondante et reprendre au point 1 avec le nouveau tableau ;

Lorsque toutes les lignes et toutes les colonnes sont saturées, le tableau doit contenir exactement (m+n-1) variables de base.

L'application à l'exemple proposé plus haut nous donne le tableau représenté dans la Figure 3.5.

	A	B	C	D	Offre
1	6	100 5	300 3	100 1	**500**
2	100 10	200 8	4	2	**300**
3	200 7	9	11	12	**200**
Demande	**300**	**300**	**300**	**100**	

Figure 3. 5 solution initiale (Coût Minimum)

Le coût total correspondant à cette solution initiale est de 5500(um).

En effet, de manière générale, la méthode du moindre coût aboutit à une meilleure solution initiale que la méthode du coin nord-ouest.

2) Recherche de la solution optimale :

Afin de trouver le tableau optimal, on procède comme dans le simplexe classique. La première étape consiste à calculer les coûts marginaux pour les variables hors-base.

Il s'agit des δ_{ij} comme indiqué dans la Figure 3.6.

	A	B	C	D	Offre
1	δ_{11} 6	$\boxed{100}$ 5	$\boxed{300}$ 3	$\boxed{100}$ 1	500
2	$\boxed{100}$ 10	$\boxed{200}$ 8	δ_{23} 4	δ_{23} 2	300
3	$\boxed{200}$ 7	δ_{32} 9	δ_{33} 11	δ_{34} 12	200
Demande	300	300	300	100	

Figure 3. 6 Les coûts marginaux.

Ce calcul se fait en trois étapes :

1. Pour chaque variable de base écrire l'équation : $c_{ij} = u_i + v_j$
2. Résoudre le système obtenu en fixant : $u_i = 0$
3. Calculer les valeurs des coûts marginaux à partir du système : $\delta_{ij} = c_{ij} - u_i - v_j$

U1=0 $\delta_{11} = c_{11} - u_1 - v_1 = 6 - 0 - 7 = -1$

$u_1 + v_2 = 5 \Rightarrow v_2 = 5$

$u_1 + v_3 = 3 \Rightarrow v_3 = 3$ $\delta_{23} = 4 - 3 - 3 = -2$

$u_1 + v_4 = 1 \Rightarrow v_4 = 4$ $\delta_{24} = 2 - 3 - 1 = -2$

$u_2 + v_2 = 8 \Rightarrow u_2 = 3$ $\delta_{32} = 9 - 0 - 5 = +4$

$u_2 + v_1 = 10 \Rightarrow v_1 = 7$ $\delta_{33} = 11 - 0 - 3 = +8$

$v_1 + u_3 = 7 \Rightarrow u_3 = 0$ $\delta_{34} = 12 - 0 - 1 = +11$

Il est pratique d'effectuer ce calcul directement sur le tableau de transport, comme la figure 3.7.

	A	B	C	D	Offre	u_i
1	-1 6	100 5	300 3	100 1	500	0
2	100 10	200 8	-2 4	-2 2	300	3
3	200 7	+4 9	+8 11	+11 12	200	0
Demande	300	300	300	100		
v_j	7	5	3	1		

Figure 3. 7 Les coûts marginaux (2)

Ce tableau contient des coûts marginaux strictement négatifs, et donc il ne s'agit pas d'une solution optimale.

– Si elle est optimale on s'arrête, sinon on refait une autre itération, et ainsi de suite.

– Ici, on peut prendre la variable x23 comme variable entrante.

– La valeur de x23 doit être augmentée mais tout en respectant les contraintes d'offre et de demande, ainsi que la non-négativité des autres variables.

Sur le tableau de la figure 3.8, le cycle indique les variations à effectuer pour sauvegarder une solution de base réalisable.

	A	B	C	D	Offre	u_i
1	-1 6	100 + 5	300 − 3	100 1	500	0
2	100 10	200 − 8	-2 + 4	-2 2	300	3
3	200 7	+4 9	+8 11	+11 12	200	0
Demande	300	300	300	100		
v_j	7	5	3	1		

Figure 3. 8 sauvegarder une solution de base réalisable.

– Un signe ``+'' indique une augmentation de la quantité et un signe ``-'' indique une diminution de la quantité.

– En valeur absolue, la variation est la même pour toutes les cases sur le cycle. Dans notre cas :

- x23 augmente de 200 unités,
- x22 diminue de 200 unités,
- x12 augmente de 200 unités
- et enfin x13diminue de 200 unités

On obtient ainsi le tableau donné par la figure 3.9, et pour lequel on refait le même travail, afin d'obtenir la solution optimale donnée par le tableau 3.10

	A	B	C	D	Offre	u_i
1	-3 + 6	300 5	100 - 3	100 1	500	0
2	100 10	+2 8	200 + 4	0 2	300	1
3	200 7	+6 9	+10 11	+12 12	200	-2
Demande	300	300	300	100		
v_j	9	5	3	1		

Figure 3. 9 Le deuxième tableau de transport ***Coût =5200 um***

	A	B	C	D	Offre	u_i
1	100 6	300 5	3	100 1	500	0
2	+3 10	+2 8	300 4	0 2	300	1
3	200 7	+3 9	+7 11	+10 12	200	1
Demande	300	300	300	100		
v_j	6	5	3	1		

Figure 3. 10 le tableau de transport (optimal)

Coût=4800 um.

Exemple 02 :

Considèrent le tableau suivant :

Destinations Origines	1	2	3	Offre
1	26	15	14	400
2	25	18	10	500
Demande	300	200	400	900

Figure 3. 11 Problème de transport (exemple 02)

❖ **Résolution du Problème**

a) Recherche de la solution réalisable de base initiale

Appliquons la méthode de Coin nord-ouest au problème de transport donné par la table :

Étape 1

Le tableau restant se compose de deux lignes et de trois colonnes ; son coin nord- ouest est la cellule (1 ; 1), x11 entre donc dans la base.

Étape 2

$$X11 = min\,\{a1;\ b1\} = min\{400;\ 300\} = 300$$

$$a'_1 = a_1 - x_{11} = 400 - 300 = 100$$

$$b'_1 = b_1 - x_{11} = 300 - 300 = 0$$

Étape 3

a'_1 = 100 et b'_1 = 0, on élimine donc la colonne 1.

Étape 4

Il nous reste deux lignes et deux colonnes, nous n'avons donc pas terminé.

Destinations Origines	~~1~~	2	3	Offre
1	~~26~~ 300	15	14	400 100
2	~~25~~	18	10	500
Demande	~~300~~ ~~0~~	200	400	900

Itération 2 :

Étape 1 :

Le tableau restant se compose des lignes 1 et 2 ainsi que des colonnes 2 et 3 ; son coin nord-ouest est donc le cellule (1 ; 2).

Étape 2 :

$$X_{12} = \min \{a_1 ; b_2\} = \min \{100 ; 200\} = 100$$

$$a'_1 = a1 - x12 = 100 - 100 = 0$$

$$b'_1 = b2 - x12 = 200 - 100 = 100$$

Étape 3 :

$a'_1 = 0$ et $b'_1 = 100$; on élimine donc la ligne 1.

Étape 4 :

Il reste une ligne et deux colonnes, il faut donc reprendre la démarche à l'étape

(1) Afin d'opérer une troisième itération.

Origines \ Destinations	~~1~~	2	3	Offre
~~1~~	~~26~~ 300	~~15~~ 100	~~14~~	~~400~~ 0
2	~~25~~	18	10	500
Demande	~~300~~ 0	200 100	300	900

Itération 3

Étape 1 :

Il ne reste que la ligne 2 et la colonne 2 et 3, donc le coin nord-ouest est la cellule (2 ;2).

Étape 2 :

$$x_{22} = \min \{a_2; b_2\} = \min\{500; 100\} = 100$$

$$a'_2 = a_2 - x_2 = 500 - 100 = 400$$

$$b'_2 = b_2 - x_{22} = 100 - 100 = 0$$

Étape 3 :

$a'_2 = 400$ et $b'_2 = 0$; on élimine donc la colonn 2.

Étape 4 :

Comme il reste une ligne et une colonne, une quatrième itération est nécessaire.

Origines \ Destinations	~~1~~	~~2~~	3	Offre
~~1~~	~~26~~ 300	~~15~~ 100	~~14~~	~~400~~ 0
2	~~25~~	~~18~~ 100	10	500 400
Demande	~~300~~ 0	~~200~~ 0	400	900

Itération 4

Étape 1 :

Il ne reste que la ligne 2 et la colonne 3, donc le coin nord-ouest est la cellule (2 ;3).

Étape 2 :

$$x_{23} = \min \{a_2; b_3\} = \min \{400; 400\} = 400$$

$$a'_2 = a_3 - x_{23} = 400 - 400 = 0$$

$$b'_2 = b_3 - x_{23} = 400 - 400 = 0$$

Étape 3 :

$a'_2 = 0$ et $b'_3 = 0$, la ligne 2 est la dernière ligne, on élimine donc la colonne 3.

Étape 4 :

Il ne reste que la ligne 2, la solution réalisable de base initiale est trouvée.

Origines \ Destinations	~~1~~	~~2~~	~~3~~	Offre
~~1~~	~~26~~ 300	~~15~~ 100	~~14~~	~~400~~ 0
~~2~~	~~25~~	~~18~~ 100	~~10~~ 400	~~500~~ 0
Demande	~~300~~ 0	~~200~~ 0	~~400~~ 0	900

-Nous pouvons calculer le coût de transport total

$$z = \sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij} x_{ij} = 26(300) + 15(100) + 18(100) + 10(400) = 15100 (\text{um}).$$

b) Recherche de la solution réalisable de base optimale

Étape 1 :

Calculons maintenant les valeurs de boucles pour les cellules hors de la base (1 ; 3) et (2 ; 1).

$$v\,(1;\,3) = c13 - c23 + c22 - c12 = 14 - 10 + 18 - 15 = 7.$$

$$v\,(2;\,1) = c21 - c11 + c12 - c22 = 25 - 26 + 15 - 18 = -4.$$

Ces valeurs sont du reste reportées dans le tableau ci-dessus, à l'endroit réservé aux x_{ij} . C'est ainsi que l'on procédera lors de tout calcul fait à la main. On remarque que $v\,(2;\,1) < 0$; par conséquent, la solution actuelle n'est pas optimale.

Origines \ Destinations	1	2	3	Offre
1	26 * 300	15 * 100	14 **(7)**	400
2	25 **(-4)**	18 * 100	10 * 400	500
Demande	300	200	400	900

Étape 4 :

On peut améliorer la solution en faisant entrer (2 ;1) dans la base. Opérons donc le changement de base tel qu'il est décrit dans la méthode d'entrée.

1. $x'_{21} = min\ \{x_{11}, x_{22}\} = min\ \{300, 100\} = 100$

2. $x'_{12} = x_{12} + x'_{21} = 100 + 100 = 200$
$x'_{11} = x_{11} - x'_{21} = 300 - 100 = 200$
$x'_{22} = x_{22} - x'_{21} = 100 - 100 = 0$

3. La cellule (2 ; 1) entre dans la base ; la cellule (2 ; 2) en sort.

Voici le tableau obtenu après le changement de base :

Origines \ Destinations	1	2	3	Offre
1	26 * 200	15 * 100	14 (7)	400
2	25 * 100	18 (4)	10 * 400	500
Demande	300	200	400	900

Le calcul des valeurs de boucle pour (1 ; 3) et (2 ; 2) nous a permis d'obtenir les résultats reportés dans le tableau. On voit que $v\ (1;\ 3)$ = 7 et $v\ (2;\ 2)$ = 4. Comme toutes deux sont non-négatives, la solution actuelle est la solution réalisable de base optimale du problème. Une remarque concernant la boucle b (1 ; 3) s'impose. La cellule (1 ; 2) ne fait en aucun cas partie de cette boucle, par conséquent, il n'y a pas trois cellules consécutives sur la ligne 1, contrairement à ce que l'on pourrait croire. Lors de la recherche d'une boucle, on peut ignorer toute cellule de base unique sur une ligne ou une colonne.

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

-Nous pouvons calculer le coût de transport total

$$z = \sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij} x_{ij} = 26(200) + 15(100) + 25(100) + 10(400) = 14700(\text{um}).$$

Conclusion

Le problème de transport est un modèle linéaire et Il ne peut pas être résolu par le simple habituel. Il existe des méthodes de résolution plus simple, non matricielle.

Dans ce chapitre nous avons essayé de résoudre le problème de transport équilibré par des différentes méthodes qui nous permettons d'obtenir une solution de base réalisable, La solution de base initiale est obtenue par la méthode Coin Nord-Ouest ou Méthode du Coût Minimum ou Méthode d'Approximation de Vogel. Ensuite les méthodes de l'optimisation de la solution de base (Stepping Stone, distribution modifiée). La méthode de Stepping Stone et la méthode de distribution modifiée peut être appliquée à n'importe quelle solution de base, elle consiste à modifier la solution de base admissible vers une solution meilleure.

Dans le chapitre suivant, nous développerons une application qui résoudre automatiquement les problèmes de transport.

Chapitre 4

Application et évaluation

Chapitre 4

Application et évaluation

Introduction

Le problème de transport est un modèle linéaire qui peut être traité par l'algorithme de résolution de problème de transport que nous avons vu au chapitre précédent.

Dans ce chapitre nous allons développer une application qui va nous permettre de résoudre des problèmes de transport. Nous utilisons le langage Java et l'outil NetBeans pour modéliser l'application.

1. Conception l'application

1.1 Méthode de conception :

Modéliser, c'est décrire de manière visuelle et graphique les besoins et, les solutions fonctionnelles et techniques d'une application. Pour ce faire, on utilisera le langage UML (Unified Modeling Language) qui nous permettra de décrire l'application sous différents angles.

Définition d'UML (Unified Modeling Language)

UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent à chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter l'application à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par l'application, etc.

Réaliser ces diagrammes revient donc à modéliser les besoins de l'application à développer.

Le langage UML fournit 13 diagrammes, qui sont classés selon deux types de vues :

- ❖ Diagrammes structurels ou diagrammes statiques (UML Structure). Parmi ces diagrammes structurels, nous avons choisi d'utiliser le diagramme de classes (Class diagram) dans notre modélisation.
- ❖ Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior). [11]

1.1.1 Diagramme de classes

La caractéristique principale d'une application réside dans sa structure. La structure de l'application est obtenue en partitionnant le domaine visé en classes et en définissant les responsabilités de chacune ainsi que les collaborations entre classes. Le noyau de notre application est constitué d'un ensemble de classes qui permettent de commencer à initialiser le problème de transport, à le résoudre avec les données d'entrée, à afficher les résultats ainsi qu'à illustrer les solutions obtenues sous forme de tableau. Les classes de notre application sont présentées à travers le diagramme de classes de la Figure 4.1. [11]

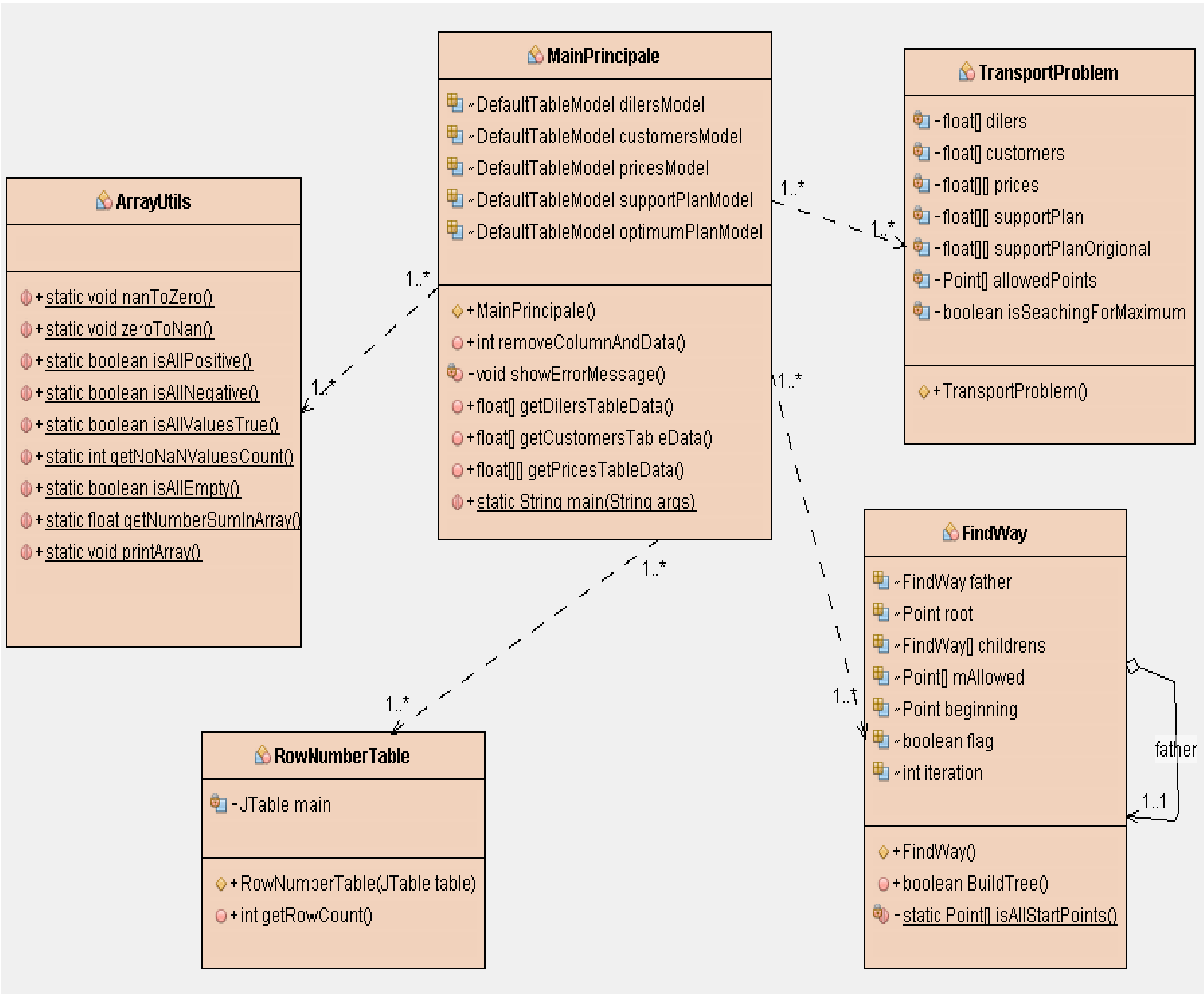

Figure 4. 1 Diagramme de classes de l'application

Une description des classes de ce diagramme est donnée ci-dessous :

❖ ***ArrayUtils :***

Opérations sur les tableaux de transport. Cette classe essaie de gérer correctement les entrées du problème de transport nulles. Une exception ne sera pas levée pour un tableau nul saisir. Cependant, un tableau Object qui contient un élément nul peut lever une exception. Chaque méthode documente son comportement.

❖ ***RowNumberTable :***

Cette classe permet ajouter ou supprimer des lignes et d'afficher l'en-tête de colonne. Pour ce faire, il ajoute le composant JTableHeader à la zone d'en-tête de colonne du JScrollPane. Bien qu'il n'y ait aucun composant dans l'API pour afficher les numéros de ligne, le volet de défilement prend également en charge la zone d'en-tête de ligne du composant d'en-tête de ligne.

❖ ***FindWay***

Cette classe représente la recherche dans le tableau de transport et Obtenir une somme donnée des éléments d'un tableau donné.

❖ ***TransportProblem***

Classe contenant les méthodes de résolution de problème de transport (Méthode de Coin Nord-Ouest, Méthode du Coût Minimum, Méthode Stepping-Stone).

❖ ***MainPrincipale***

Cette classe exécute les méthodes pour obtenir la solution du problème de transport obtenu à partir de la classe TransportProblem.

2. Implémentation

2.1 Description de l'environnement de développement

2.1.1 Langage JAVA :

Java est un langage de programmation orienté objet développé par Sun Microsystems (aujourd'hui racheté par Oracle).

Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers une représentation binaire intermédiaire qui peut être exécutée dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation. Nous avons implémenté notre application avec ce langage en utilisant l'IDE NetBeans. [12]

2.1.2 L'IDE NetBeans

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java). L'IDE NetBeans s'appuie sur cette plateforme.

L'environnement de base comprend les fonctions générales suivantes :

Configuration et gestion de l'interface graphique des utilisateurs,

Support de différents langages de programmation,

Traitement du code source (édition, navigation, formatage, inspection ...),

Fonctions d'import/export depuis et vers d'autres IDE, tels qu'Eclipse ou JBuilder,

Accès et gestion de bases de données, serveurs Web, ressources partagées,

Gestion de tâches (à faire, suivi ...), Documentation intégrée. [13]

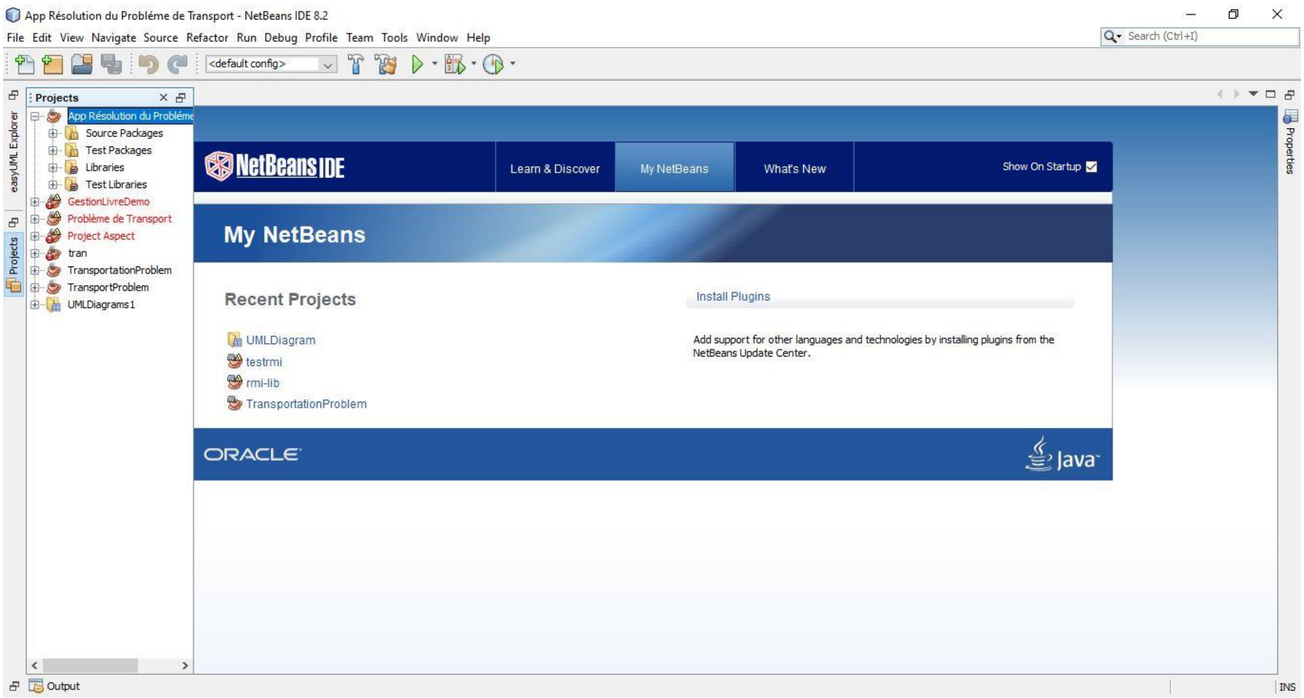

Figure 4. 2 Interface de NetBeans

2.2 Présentation de l'application

Maintenant, on va présenter les interfaces de l'application, et les expliquer.

L'interface Principale :

Lorsque l'application est exécutée, cette interface apparaît qui contient les informations de l'application. **(Figure 4. 3)**

Figure 4. 4 l'interface Principale d'application.

L'interface résoudre le problème de transport

Cette interface comporte deux sections, une page pour saisir les données du problème de transport et une page pour résoudre et afficher le problème. (**Figure 4. 5)**

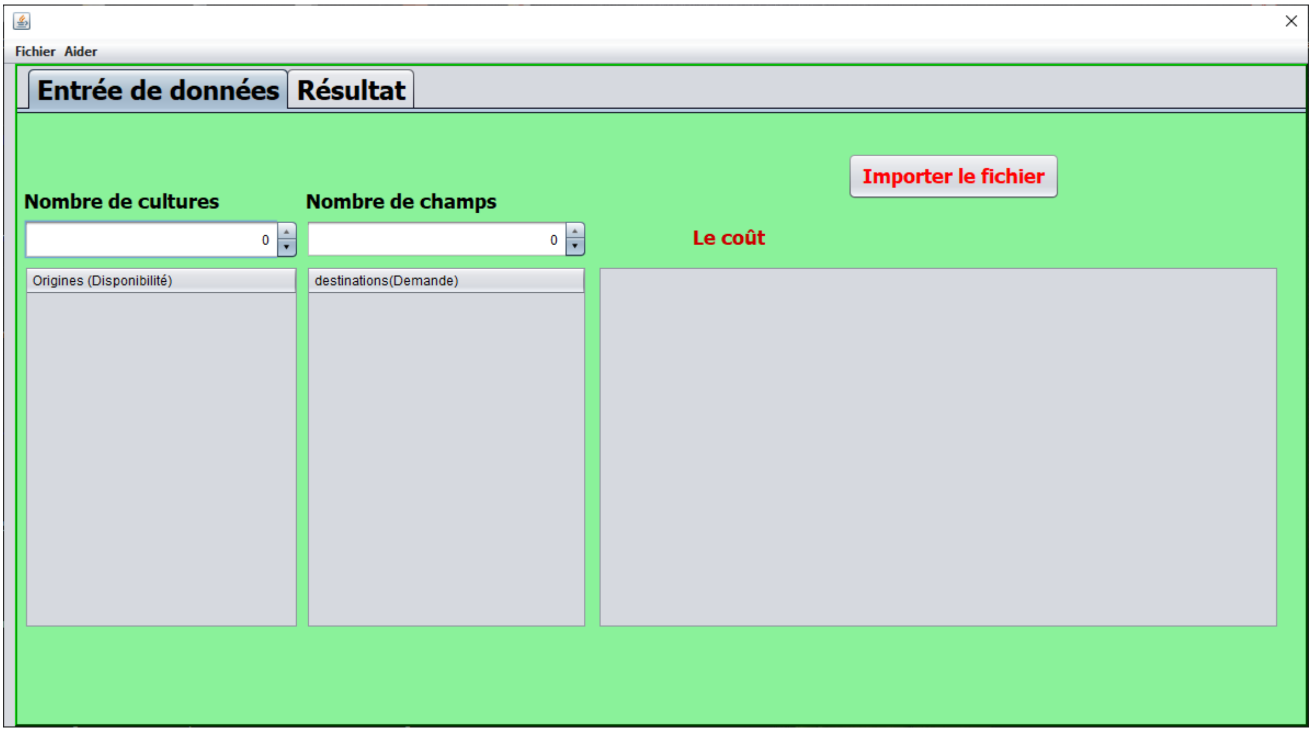

Figure 4. 6 L'interface résoudre le problème de transport.

Page Entrée de données

Sur cette page, le tableau de transport est dessiné et les données y sont saisies de manière simple, ou nous pouvons importer le fichier de données et le tableau de transport est dessiné automatiquement.

Figure 4. 7 Dessiné le tableau de transport

- ***Importer le Fichier***

Lorsque l'utilisateur appuie sur le bouton « Importer le Fichier » la fenêtre suivante s'affiche, puis on lit les données qui sont déjà insérées sur un fichier document texte. (**Figure 4. 8)**

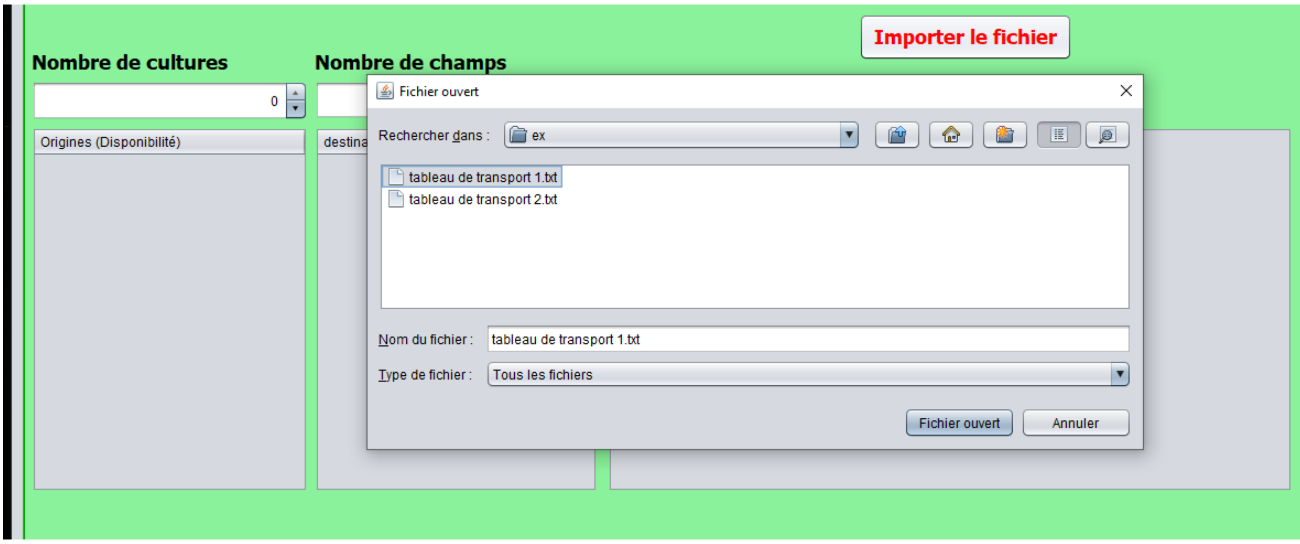

Figure 4. 8 Fenêtre d'accès aux fichiers

Page Résultat

Cette page résout le problème de transport donné et affiche le résultat.

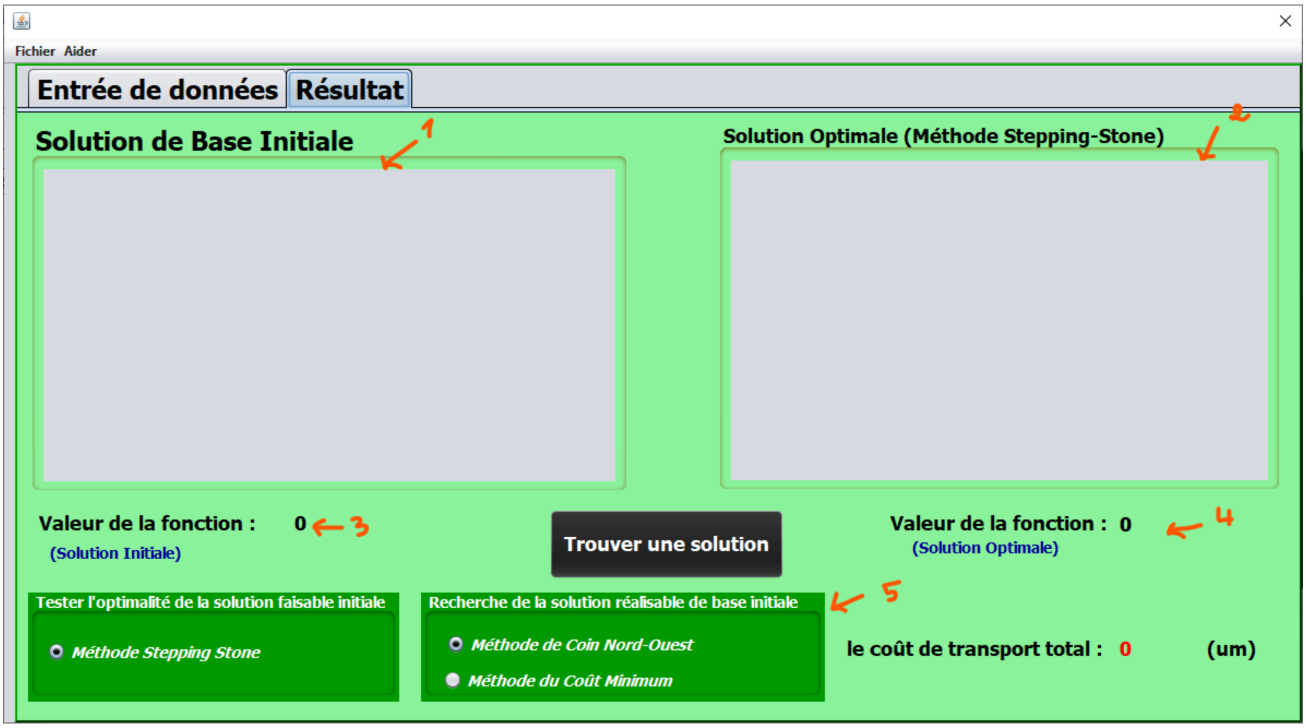

Figure 4. 9 Fenêtre de résultat

1- Dans cette fenêtre, un tableau apparaît pour la solution réalisable initiale au problème de transport.

2- Dans cette fenêtre, un tableau apparaît pour la solution optimale au problème de transport.

3- Ici apparaît la valeur du coût de transport total obtenu par la méthode de Coin Nord-Ouest ou Méthode du Coût Minimum.

4- Ici apparaît la valeur du coût de transport total obtenu par la Méthode Stepping-Stone.

5- Choisir une méthode pour une solution initiale réalisable au problème de transport.

2.3 Exemple d'une solution de problème de transport par l'application

	Dest 1	*Dest 2*	*Dest 3*	*Dest 4*	*Dest 5*	*Dest 6*	***Disponibilité***
O_1	40	30	18	24	16	9	**600**
O_2	45	28	22	18	22	17	**180**
O_3	50	22	14	16	12	7	**220**
O_4	38	20	12	12	18	12	**300**
O_5	56	18	19	20	14	10	**195**
O_6	42	32	20	18	24	17	**290**
O_7	35	26	18	26	26	14	**400**
O_8	28	24	27	24	20	16	**500**
Demande	**270**	**800**	**110**	**140**	**400**	**260**	

Figure 4. 10 Exemple tableau de transport

1. Insérer les données dans le Bloc-notes dans l'ordre, comme indiqué sur la figure 4.11

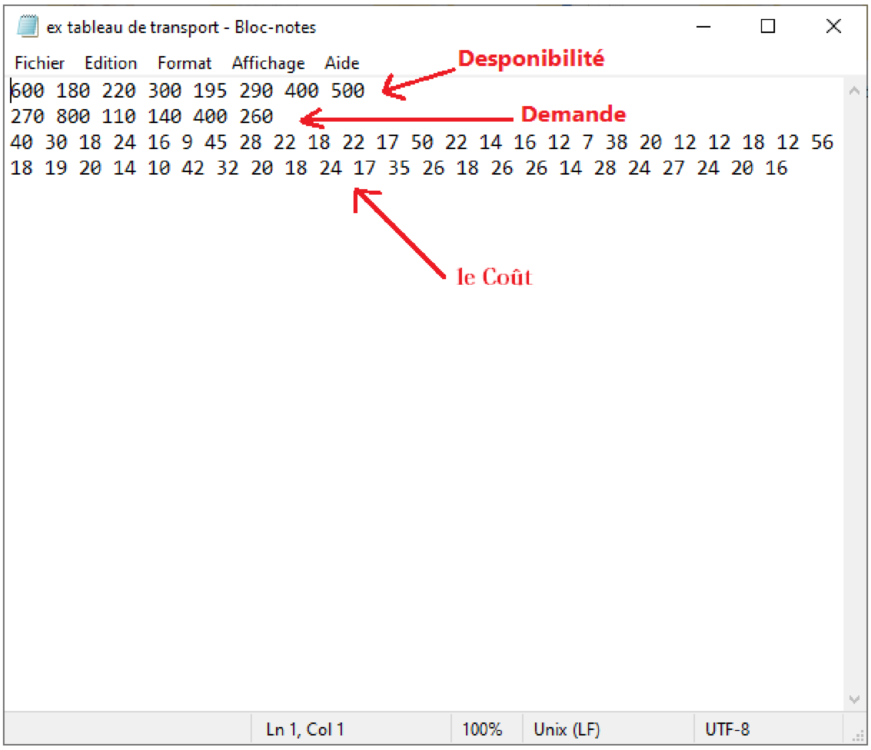

Figure 4. 11 fichier de données

2. Parcourir le fichier des données :

Cliquer sur le bouton "Importer le fichier" pour importer les données et lancer la résolution du problème.

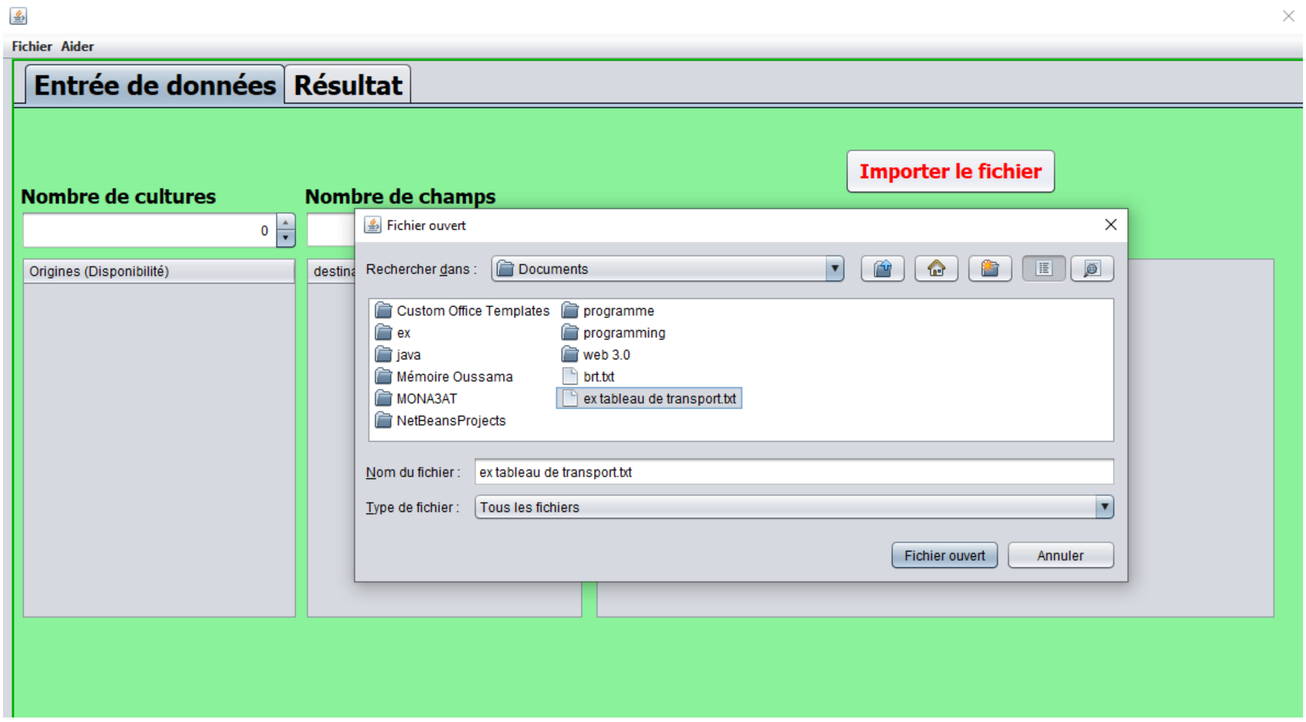

Figure 4. 12 Ecran de parcours du fichier

3. Charger le fichier de données :

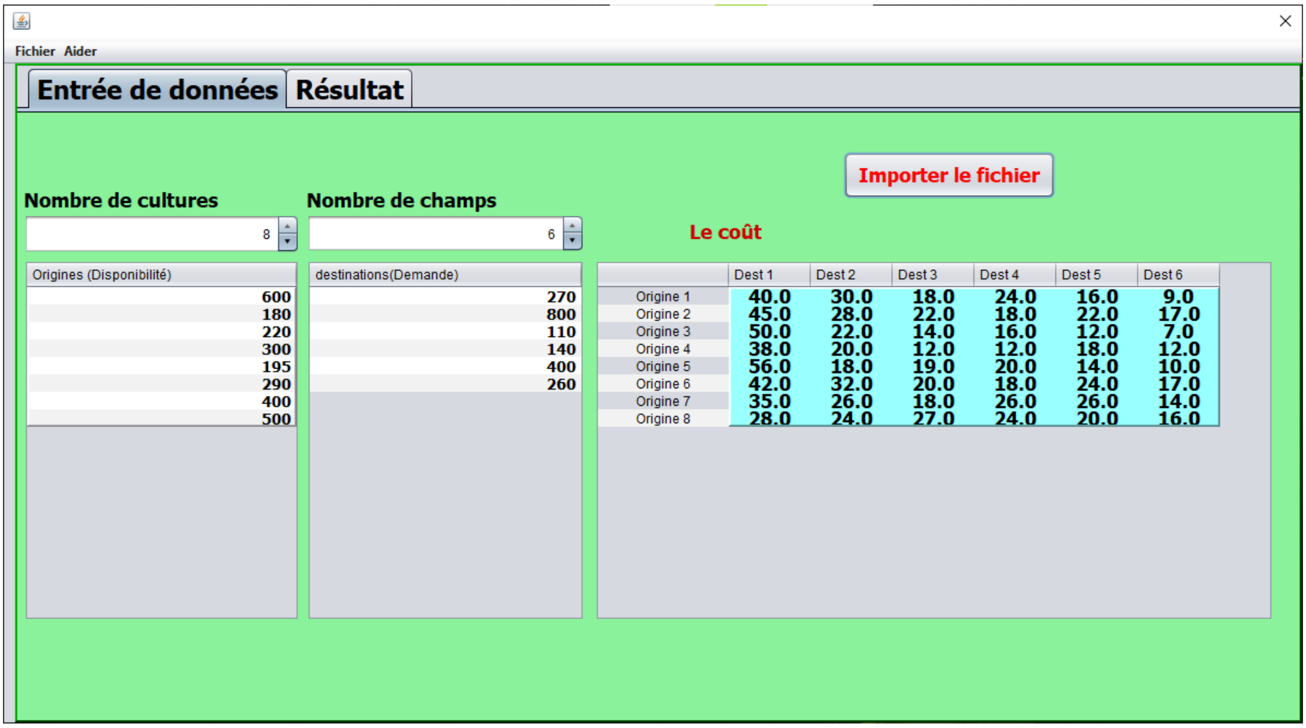

Figure 4. 13 Ecran de chargement des données

4. Résolution du problème

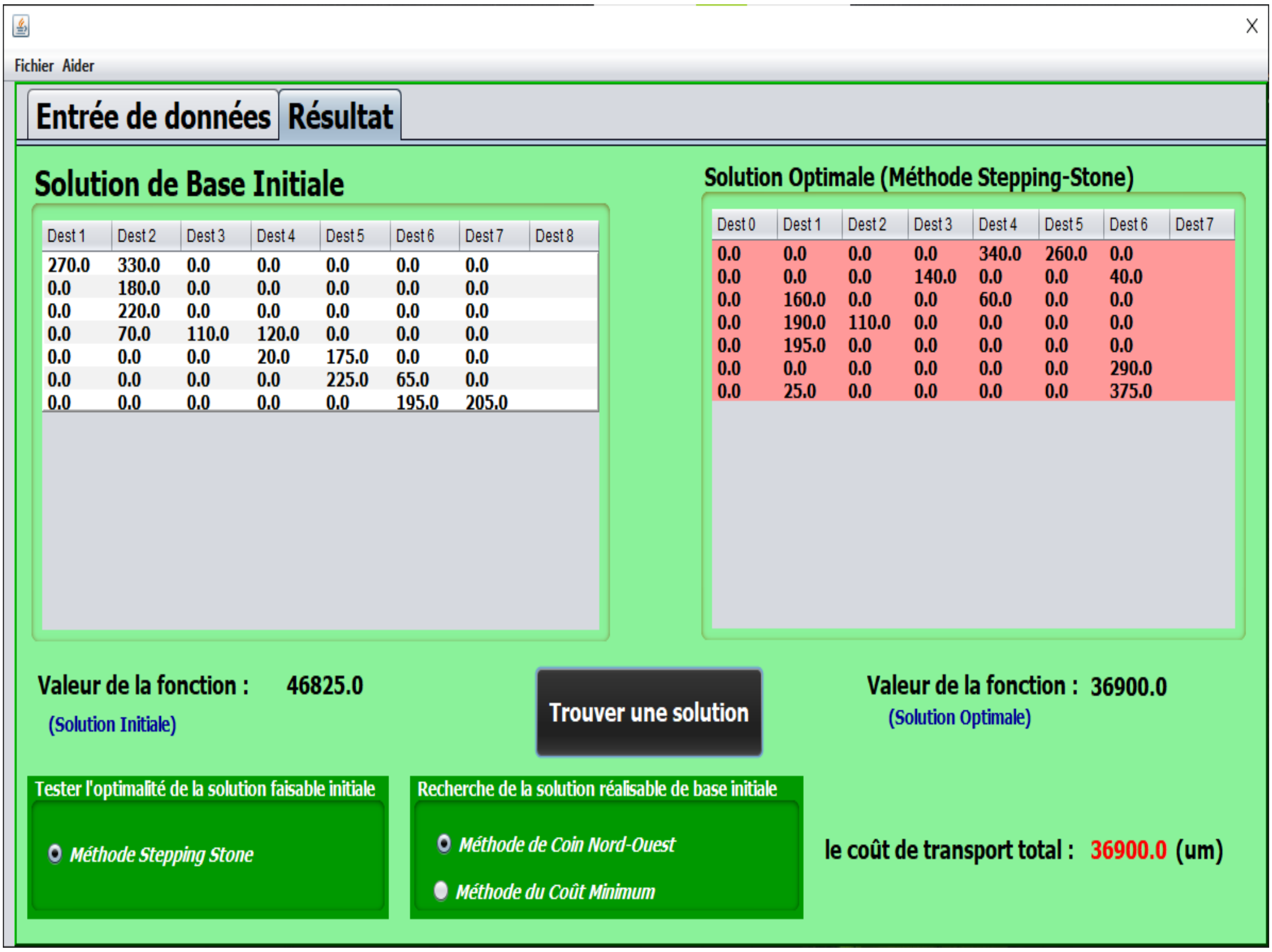

Figure 4. 14 Écran visualisant la solution finale

❖ Méthode de Coin Nord-Ouest

- Cet écran visualise la solution de base et le coût total obtenus par la méthode du Coin Nord-Ouest (**Figure 4.16**) (**Figure 4.17**) :

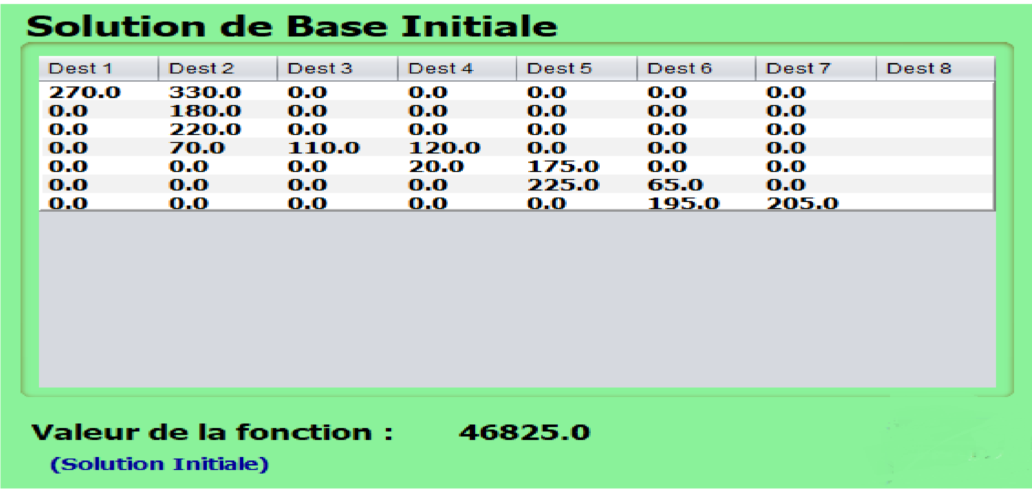

Figure 4. 15 Ecran visualisant la solution par Méthode du Coin Nord-Ouest (1)

```
run:
Origine (Disponibilité)
[600.0, 180.0, 220.0, 300.0, 195.0, 290.0, 400.0, 500.0]
Destination (demande)
[270.0, 800.0, 110.0, 140.0, 400.0, 260.0, 705.0]
le coût
40.0    30.0    18.0    24.0    16.0    9.0     0.0
45.0    28.0    22.0    18.0    22.0    17.0    0.0
50.0    22.0    14.0    16.0    12.0    7.0     0.0
38.0    20.0    12.0    12.0    18.0    12.0    0.0
56.0    18.0    19.0    20.0    14.0    10.0    0.0
42.0    32.0    20.0    18.0    24.0    17.0    0.0
35.0    26.0    18.0    26.0    26.0    14.0    0.0
28.0    24.0    27.0    24.0    20.0    16.0    0.0
Plan de départ
270.0   330.0   NaN     NaN     NaN     NaN     NaN
NaN     180.0   NaN     NaN     NaN     NaN     NaN
NaN     220.0   NaN     NaN     NaN     NaN     NaN
NaN     70.0    110.0   120.0   NaN     NaN     NaN
NaN     NaN     NaN     20.0    175.0   NaN     NaN
NaN     NaN     NaN     NaN     225.0   65.0    NaN
NaN     NaN     NaN     NaN     NaN     195.0   205.0
NaN     NaN     NaN     NaN     NaN     NaN     500.0
Coût Total =  46825.0
m + n - 1: 14
Nombre de cellules de base : 14
```

Figure 4. 16 Ecran visualisant la solution par Méthode du Coin Nord-Ouest (2)

❖ Méthode du Coût Minimum

- Cet écran visualise la solution faisable initiale et le coût total obtenus par la Méthode du Coût Minimum **(Figure 4.19)** :

Solution de Base Initiale

Dest 1	Dest 2	Dest 3	Dest 4	Dest 5	Dest 6	Dest 7	Dest 8
120.0	0.0	0.0	0.0	205.0	40.0	235.0	
0.0	0.0	0.0	0.0	0.0	0.0	180.0	
0.0	0.0	0.0	0.0	0.0	220.0	0.0	
0.0	50.0	110.0	140.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	195.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	290.0	
150.0	250.0	0.0	0.0	0.0	0.0	0.0	

Valeur de la fonction : 40460.0
(Solution Initiale)

Figure 4. 17 Ecran visualisant la solution par Méthode du Coût Minimum(1)

```
run:
Origine (Disponibilité)
[600.0, 180.0, 220.0, 300.0, 195.0, 290.0, 400.0, 500.0]
Destination (demande)
[270.0, 800.0, 110.0, 140.0, 400.0, 260.0, 705.0]
le coût
40.0    30.0    18.0    24.0    16.0    9.0     0.0
45.0    28.0    22.0    18.0    22.0    17.0    0.0
50.0    22.0    14.0    16.0    12.0    7.0     0.0
38.0    20.0    12.0    12.0    18.0    12.0    0.0
56.0    18.0    19.0    20.0    14.0    10.0    0.0
42.0    32.0    20.0    18.0    24.0    17.0    0.0
35.0    26.0    18.0    26.0    26.0    14.0    0.0
28.0    24.0    27.0    24.0    20.0    16.0    0.0
Plan de départ
120.0   NaN     NaN     NaN     205.0   40.0    235.0
NaN     NaN     NaN     NaN     NaN     NaN     180.0
NaN     NaN     NaN     NaN     NaN     220.0   NaN
NaN     50.0    110.0   140.0   NaN     NaN     NaN
NaN     NaN     NaN     NaN     195.0   NaN     NaN
NaN     NaN     NaN     NaN     NaN     NaN     290.0
150.0   250.0   NaN     NaN     NaN     NaN     NaN
NaN     500.0   NaN     NaN     NaN     NaN     NaN
Coût Total =  40460.0
m + n - 1: 14
Nombre de cellules de base : 14
```

Figure 4. 18 Ecran visualisant la solution par Méthode du Coût Minimum (2)

❖ Méthode du Stepping Stone

Cet écran visualise la solution optimale et le coût total obtenus par la méthode du Stepping Stone.

Solution Optimale (Méthode Stepping-Stone)

Dest 0	Dest 1	Dest 2	Dest 3	Dest 4	Dest 5	Dest 6	Dest 7
0.0	0.0	0.0	0.0	340.0	260.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	180.0	
0.0	160.0	0.0	0.0	60.0	0.0	0.0	
0.0	50.0	110.0	140.0	0.0	0.0	0.0	
0.0	195.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	290.0	
0.0	165.0	0.0	0.0	0.0	0.0	235.0	

Valeur de la fonction : 36900.0
(Solution Optimale)

le coût de transport total : 36900.0 (um)

Figure 4. 19 Ecran visualisant la solution par Méthode de Stepping Stone

2.4 Complexité algorithmique :

- **Coin du Nord-Ouest :**

Certainement la méthode la plus facile à mettre en œuvre et de complexité minimale, seulement en O(m+n). Elle consiste à déterminer la variable de plus petits indices où il est possible d'affecter une quantité. Sa simplicité l'a rendue très populaire. Par contre, la solution trouvée n'a aucune raison d'être près de la solution optimale.

- **Coût minimum**

La méthode est facile à mettre en œuvre, elle trouve des solution de départ proche à la solution optimal, cette méthode est de complexité O((m + n)2) , la méthode consiste à déterminer le coût le plus petits de transporter une quantité.

2.4.1 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, on a résumé les résultats des coûts de transport dans le tableau suivant :

Problème :	Taille de problème	Coin Nord-Ouest	Coût minimum		Optimisation (stepping stone)
1	3x4	3860	3670		3460
2	3x3	6600	6460		5920
3	3x3	730	555		555
4	3x5	363	305		290
5	3x4	176	176		149
6	3x3	5925	4550		4525
7	3x4	7750	7650		7225
8	3x4	310	180		180
9	3x4	670	670		610
10	3x3	175	148		148

Comparaison :

Nous avons utilisé ici trois méthodes : Méthode de coin Nord-ouest, méthode de coût Minimum pour trouver une première solution possible de base pour le modèle de transport. Et la méthode d'optimisation de la solution de base (stepping stone). Le coût du transport montre que :

-Dans la méthode de coût Minimum, la majorité des résultats sont meilleurs (optimisation) que la méthode coin du Nord-Ouest.

-A titre d'exemple le problème de transport numéro 8 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 310, Or avec la méthode de coût minimum, la solution de base obtenu est de coût total inférieur à celui de Nord-ouest égale à 180 est le plus proche d'optimum et la solution optimale est égale 180.

Pour les problèmes de plus grande taille, nous remarquons que aussi la méthode de coût Minimum semble être à efficace. Nous en déduisons que de ce point de vue la méthode de coût Minimum est meilleure que la méthode de Coin Nord-Ouest .

Conclusion

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 de coût Minimum 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 JAVA comme un programme de résolution d'un problème de transport.

Dans un prochain avenir nous espérons pouvoir continuer à travailler sur les problèmes de la recherche opérationnelle ainsi que leur résolution, car c'est un domaine vaste, riche et très intéressant.

Bibliographie

[1] N. Kahoul, Optimisation Linéaire, Ezzouar, 2019.

[2] B. Fortz, Recherche opérationnelle et applications, Bruxelles, 2012.

[3] R. J. Vanderbei, LINEAR PROGRAMMING Foundations and Extensions, New Jersey, USA: Springer, 2008.

[4] D. Werra, Operations Research (STA 2209), Kenyatta, 2021.

[5] D. YADOLAH, Optimisation appliquée, Etat de l'Utah: Springer Editions,, 2005.

[6] J. Y. WANG, Operation Research I, College of Management NCTU, 2008.

[7] W. a. M. V. L.WAYNE, Mathematical Programming : Operations Research, Volume 1 4eme édition, 2003.

[8] R. O. R. P. Yves Nobert, Méthodes de planification en transport, 2014.

[9] P. R. Murthy, Operations researc, New Age International (P) Ltd, 2007.

[10] M. SIMONNARD, Linear Programming, 1966.

[11] T. Contributor, «techtarget,» 2010. [En ligne]. [Accès le 2022].

[12] J. Hartman, «guru99,» 2022. [En ligne]. [Accès le 2022].

[13] «techno-science,» 2022. [En ligne].

[14] J. Y. WANG, Operation Research I, College of Management NCTU, 2008.