

## CHAPITRE 2

### Le problème de transport

#### 1. Introduction :

Les problèmes de transport jouent un rôle très important dans de nombreux secteurs d'activités. On peut considérer que résoudre un problème de transport consiste à organiser dans le temps et dans l'espace un ensemble de tournées affectées à divers véhicules. Ce type de problème est rencontré, par exemple, dans les chaînes logistiques de grands groupes industriels, dans la grande distribution, dans les systèmes de transports publics (métro, bus...) ou privés (Uber, Blablacar...). Les problèmes se rencontrent aussi bien dans le cadre des services à la personne (transport de personnes à mobilité réduite, transport de personnes âgées, transport scolaire...) que dans le cadre du transport de marchandises (livraison de colis, collecte de déchets ménagers...). Les problèmes de tournées sont des problèmes combinatoires, c'est-à-dire des problèmes pour lesquels il existe un grand nombre de solutions qu'il est, en général, impossible de parcourir en totalité. Au cours de cette thèse, plusieurs problèmes de tournées sont traités, avec des approches essentiellement heuristiques. Ces méthodes ne garantissent pas l'optimalité de la solution mais permettent de fournir une solution de bonne qualité avec des temps de calcul raisonnables.

#### 2. Le problème de transport :

Un problème de transport peut être défini comme l'action de transporter depuis "m origines" vers "n destinations" des matériaux, au moindre coût. Donc, la résolution d'un problème de transport consiste à organiser le transport de façon à minimiser son coût.

Formulation :

$a_i$  = production ou offre       $a_i \in N \quad \forall i \in [1, \dots, m]$

$b_j$  = demande       $b_j \in N \quad \forall j \in [1, \dots, n]$

$x_{ij}$  = quantité transportée       $x_{ij} \in N \quad \forall i \in [1, \dots, m]; \quad \forall j \in [1, \dots, n]$

$c_{ij}$  = Le cout unitaire de transport de l'origine [i] vers la destination [j]       $c_{ij} \in N$ .

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \quad \forall i \in [1, \dots, m]; \quad \forall j \in [1, \dots, n]; \quad (1)$$

$$\sum_{j=1}^n x_{ij} = a_i \quad \forall i \in [1, \dots, m]; \quad (2)$$

$$\sum_{i=1}^m x_{ij} = b_j \quad \forall j \in [1, \dots, n]; \quad (3)$$

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

Exemple :

Soit, une société possédant quatre dépôts A1, A2, A3 et A4 dans lesquels existent des quantités respectives de 150, 350, 450, 550 unités d'une matière première, et cinq usines D1, D2, D3, D4 et D5 demandant respectivement 100, 200, 300, 400 et 500 unités de celles-ci. Les coûts de transport,  $C_{ij}$ , sont donnés par le tableau ci-dessous. Comment organiser le transport au moindre coût total?

	D1	D2	D2	D4	D5	$a_i$
A1	73	65	19	14	48	150
A2	61	13	21	65	41	350
A3	90	84	88	94	29	450
A4	10	14	93	17	16	550
$b_j$	100	200	300	400	500	1500

**Table2.1** : cout unitaire de « Les origines vers les destinations ».

## 2.1 Méthode de résolution: recherche d'une solution de base réalisable :

Solution de base :

On appelle solution de base d'un programme de transport, une solution admissible comportant  $M = (m+n-1) x_{ij} > 0$ , c'est-à-dire qu'une solution de base comporte  $(m.n - M)$  zéros. Le graphe d'une solution de base est un graphe connexe sans cycle, c'est-à-dire un arbre comportant  $N=m+n$  sommets soit  $M=N-1$  arcs.[12]

On distingue trois méthodes connues dans la littérature de détermination de solution de base réalisable (approchée).

- La méthode du cout minimal (CM)
- La méthode du Vogel (MV)
- La méthode du coin Nord-Ouest (NO)

### 2.1.1 Méthode du cout minimal (CM) :

Cette méthode tire son nom de la priorité qu'elle accorde à l'acheminement de quantités les plus grandes possible par des routes origine-destination dont les coûts unitaires de transport sont les plus faibles. Acheminer une quantité maximale de biens sur une telle route  $i$ - $j$  correspond à attribuer la plus grande valeur possible à une case  $(i, j)$  choisie parmi celles des cases disponibles dont le coût unitaire de transport est minimal. Une est dite disponible tant qu'elle n'a pas fait l'objet d'un choix ou n'a pas été «éliminée» à la suite d'un choix. Appliquons cette méthode au problème de transport de Pourceau.

Principe :

1. le choix d'une case disponible de coût unitaire minimal (s'il y a plusieurs candidates, on cherche à maximiser la valeur qui sera attribuée lors de l'étape 2 ; s'il en reste encore plus d'une, on choisit de façon aléatoire parmi les cases candidates restantes);
2. l'attribution à la case  $(i, j)$  retenue du nombre le plus grand possible, compte tenu des Disponibilités et des demandes résiduelles indiquées au tableau;
3. la mise à jour de la demande non comblée de  $C_j$  et de la disponibilité résiduelle de  $L_i$  ;
4. la saturation d'une rangée (à choisir entre la ligne  $L_i$  et la colonne  $C_j$ ) dont la marge Contient maintenant le nombre 0 ; l'inscription de tirets dans les cases de la rangée déclarée saturée qui étaient encore disponibles. [16]

#### Algorithme du CM :

**Début :**

**Répéter !** (disponibilité = restriction = 0)

- $\text{Min } C_{ij} \leftarrow \text{Min (Restriction } j ; \text{Disponibilité } i) ;$
- $\text{Disponibilité } i \leftarrow | \text{Disponibilité } i - \text{Min } | ;$
- $\text{Restriction } j \leftarrow | \text{Restriction } j - \text{Min } | ;$
- Si  $\text{Disponibilité } i = 0$  supprimer la colonne  $i$  ;
- Si  $\text{Restriction } j = 0$  supprimer la ligne  $j$  ;

**FIN Répéter**

Retourner la fonction objective ( $f^*$ )  $\{ \text{cout unitaire } C_{ij} * \text{unités } C_{ij} \}$

**Fin Début**

**Algorithme 2.1.** pseudo-code de CM

Exemple :

Appliquer la méthode du cout minimal pour déterminer une solution approchée pour le problème de transport suivant :

Première étape :

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER	8	5	6	120
M'SILA	15	10	12	80
SETIF	3	9	10	80
Demande	150	70	60	280

	ORAN	ANNABA	BISKRA
ALGER			
M'SILA			
SETIF	80		

Deuxième étape :

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER	8	5	6	120
M'SILA	15	10	12	80
Demande	70	70	60	200

	ORAN	ANNABA	BISKRA
ALGER		70	
M'SILA			
SETIF	80		

Troisième étape :

	ORAN	BISKRA	Disponibilité
ALGER	8	6	50
M'SILA	15	12	80
Demande	70	60	130

	ORAN	ANNABA	BISKRA
ALGER		70	50
M'SILA			
SETIF	80		

Quatrième étape :

	ORAN	BISKRA	Disponibilité
M'SILA	15	12	80
Demande	70	10	80

	ORAN	ANNABA	BISKRA
ALGER		70	50
M'SILA			10
SETIF	80		

Dernière étape :

	ORAN	Disponibilité
M'SILA	15	70
Demande	70	70

	ORAN	ANNABA	BISKRA
ALGER		70	50
M'SILA	70		10
SETIF	80		

Enfin, la solution de base est :

	ORAN	ANNABA	BISKRA
ALGER	8	5	6
M'SILA	15	10	12
SETIF	3	9	10

Son coût est  $(80 \times 3) + (70 \times 15) + (70 \times 5) + (50 \times 6) + (10 \times 12) = 2060$

### 2.1.2 Méthode de Vogel (MV) :

Cette méthode est basée sur le calcul des regrets. Le regret associé à une ligne ou à une colonne est la différence entre le coût minimum et le coût immédiatement supérieur dans cette ligne ou dans cette colonne. C'est une mesure de la priorité à accorder aux transports de cette ligne ou de cette colonne, car un regret important correspond à une pénalisation importante si on n'utilise pas la route de coût minimum.

La méthode de Vogel approximation fournit, en général, une solution très proche de l'optimum; le nombre de changements de base nécessaires pour arriver à une solution optimale est peu élevé (il arrive même assez fréquemment que la solution donnée par cette règle soit optimale).

Principe:

D'abord, on calcule pour chaque rangée, ligne ou colonne, la différence entre le coût le plus petit avec celui qui lui est immédiatement supérieur. Ensuite on affecte à la relation de coût le plus petit correspondant à la rangée présentant la différence maximale la quantité la plus

élevée possible. Ce qui sature une ligne ou une colonne. Et on reprendre le processus jusqu'à ce que toutes les rangées soient saturées.

$\Delta l$  représente la différence entre le coût minimum et celui immédiatement supérieur sur une ligne.

$\Delta c$  représente la différence entre le coût minimum et celui immédiatement supérieur sur une colonne.

- 1- Calculer les différences  $\Delta l$  et  $\Delta c$  pour chaque ligne et colonne.
- 2- Sélectionner la ligne ou la colonne ayant le  $\Delta l$  ou  $\Delta c$  maximum.
- 3- Choisir dans cette ligne ou colonne le coût le plus faible.
- 4- Attribuer à la relation  $(i, j)$  correspondante le maximum possible de matière transportable de façon à saturer soit la destination soit la disponibilité.
- 5- Calculer la quantité résiduelle soit en demande soit en disponibilité.
- 6- Eliminer la ligne ou la colonne ayant sa disponibilité ou demande satisfaite.
- 7- Si le nombre de lignes ou colonnes  $> 2$  retour en 2. SINON affecter les quantités restantes aux liaisons. [12]

**Algorithme MV :**

**Début :**

- **Répéter !** (Disponibilité = Restriction = 0)
- $P \leftarrow$  Calculate penalty value ;
  - $(i \text{ or } j) \leftarrow \text{Max } (P)$ . Cet indice;
  - $(i, j) \leftarrow \text{Min } C_{ij}$  . Cet indice;
- $C_{ij} \leftarrow \text{Min } (\text{Restriction } j ; \text{Disponibilité } i) ;$
- $\text{Restriction } j \leftarrow \text{Restriction } j - C_{ij} ;$
- $\text{Disponibilité } i \leftarrow \text{Disponibilité } i - C_{ij} ;$
- Si  $(\text{Restriction } j == 0)$  Supprimer Colonne ;
- Si  $(\text{Disponibilité } i == 0)$  Supprimer ligne ;

**Fin Répéter**

Retourner tableau ;

Retourner fect obj {Cout unitaire  $ij$  \* unité  $ij$  } ; **Fin Début**

**Algorithme 2.2.** pseudo-code de MV.

Exemple :

Appliquer la méthode de Vogel pour déterminer une solution approchée pour le problème de transport suivant :

Première étape :

	ORAN	ANNABA	BISKRA	Disponibilité	$\Delta I$
ALGER	8	5	6	120	1
M'SILA	15	10	12	80	2
SETIF	3	9	10	80	6
Demande	150	70	60	280	
$\Delta c$	5	4	4		

Deuxième étape

	ORAN	ANNABA	BISKRA
ALGER			
M'SILA			
SETIF	80		

	ORAN	ANNABA	BISKRA	Disponibilité	$\Delta I$
ALGER	8	5	6	120	1
M'SILA	15	10	12	80	2
Demande	70	70	60	200	
$\Delta c$	7	5	6		

	ORAN	ANNABA	BISKRA
ALGER	70		
M'SILA			
SETIF	80		

Troisième étape :

	ANNABA	BISKRA	Disponibilité	$\Delta I$
ALGER	5	6	50	1
M'SILA	10	12	80	2
Demande	70	60	130	
$\Delta c$	5	6		

	ORAN	ANNABA	BISKRA
ALGER	70		50
M'SILA			
SETIF	80		

Quatrième étape :

	ANNABA	BISKRA	Disponibilité	$\Delta I$
M'SILA	10	12	80	2
Demande	70	10	80	
$\Delta c$	0	0		

	ORAN	ANNABA	BISKRA
ALGER	70		50
M'SILA		70	
SETIF	80		

Dernière étape :

	BISKRA	Disponibilité	$\Delta I$
M'SILA	12	10	2
Demande	10	10	
$\Delta c$	0		

Enfin, la solution de base est :

	ORAN	ANNABA	BISKRA
ALGER	70		50
M'SILA		70	10
SETIF	80		

	ORAN	ANNABA	BISKRA
ALGER	8	5	6
M'SILA	15	10	12
SETIF	3	9	10

Son coût est  $= (80 \times 3) + (70 \times 8) + (70 \times 10) + (50 \times 6) + (10 \times 12) = 1920$

### 2.1.3 Méthode du coin Nord-Ouest : (NO)

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 raison la que généralement la solution obtenue par cette méthode est loin de la solution optimale.

Principe :

On considère à chaque étape, la case la plus Nord à l'Ouest de la matrice des coûts. On part donc de la route  $(i1 ; j1)$  ; on sature soit la ligne  $i1$  soit la colonne  $j1$ . Puis on recommence sur la sous-grille formée des lignes et des colonnes non saturées. [12]

Donc l'idée de la méthode est de remplir au maximum la case du tableau en haut, à gauche, puis compléter sur la ligne ou la colonne (de façon à atteindre l'offre ou la demande) et continuer ainsi à compléter les cases immédiatement à droite et en dessous alternativement.

On peut résumer la méthode dans l'algorithme suivant : 1-  $I=1, j=1$

2-  $C_{ij} = \min(a_i; b_j)$ . Si  $C_{ij} = a_i$  passé à (3) sinon passer à (4).

3- Poser  $b_j = b_j - a_i$ ; et  $i=i+1$ , si  $i \leq n$  passer à (2) sinon fin.



4- Poser  $a_i = a_i - b_j$  et  $j=j+1$ , si  $j \leq m$  passer à (2) sinon fin.

**Algorithme de NO:**

**Début :**

**Répéter !** (disponibilité = restriction = 0)

- $i, j \leftarrow$  aller vers Nord-Ouest disponible;
- $C_{ij} \leftarrow \text{Min}(\text{Restriction } j; \text{Disponibilité } i);$
- $\text{Restriction } j \leftarrow \text{Restriction } j - C_{ij};$
- $\text{Disponibilité } i \leftarrow \text{Disponibilité } i - C_{ij};$
- Si ( $\text{Restriction } j == 0$ ) supprimer colonne  $j$  ;
- Si ( $\text{Disponibilité } i == 0$ ) supprimer ligne  $i$  ;

**Fin Répéter**

Retourne tableau ;

Retourne fct obj {cout unitaire  $ij$  \* unité  $ij$  } ;

**Fin début**

**Algorithme 2. 3.** pseudo-code de NO.

Pour bien comprendre l'idée de la méthode on va l'appliquer sur l'exemple précédant :

<b>NO</b>	ORAN	ANNABA	BISKRA	Disponibilité
ALGER	8	5	6	120
M'SILA	15	10	12	80
SETIF	3	9	10	80
Demande	150	70	60	280

Dans le coin Nord-Ouest on met la plus grande valeur qui va soit satisfaire une demande ou bien épuiser une disponibilité. Dans l'exemple on va épuiser la 1ère disponibilité.

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER	120			120-120=0
M'SILA				80
SETIF				80
Demande	150-120=30	70	60	180

Ensuite on refait la même chose mais sur le nouveau sous tableau qu'on obtient après la 1ère

modification	ORAN	ANNABA	BISKRA	Disponibilité
ALGER	120			0
M'SILA	30			80-30=50
SETIF				80
Demande	30-30=0	70	60	130

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER	120			0
M'SILA	30	50		0
SETIF		20	→	80-20=60
Demande	0	20-20=0	60	60

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER	120			0
M'SILA	30	50		0
SETIF		20	60	60-60=0
Demande	0	0	60-60=0	0

A la fin on obtient notre solution de base qui vérifie les contraintes de problème, donc la solution de base ainsi obtenue est représenté dans le tableau suivant :

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER	120			120
M'SILA	30	50		80
SETIF		20	60	80
Demande	150	70	60	280

Le coût total de cette solution :

$$Z = (120*8) + (30*15) + (50*10) + (20*9) + (60*10) = 2690$$

#### 2.1.4 Nouvelle Heuristique : Nord-Est (NE) :

Ceci est inspiré de la méthode nord-est. Mais on allant du coin nord-est en utilisant un mécanisme similaire.

Principe :

On considère à chaque étape, la case la plus Nord à l'Est de la matrice des coûts. On part donc de la route (i1 ; j m) ; on sature soit la ligne i1 soit la colonne j m. Puis on recommence sur la sous-grille formée des lignes et des colonnes non saturées.

Donc L'idée de la méthode est de remplir au maximum la case du tableau en haut, à droite, puis compléter sur la ligne ou la colonne (de façon à atteindre l'offre ou la demande) et continuer ainsi à compléter les cases immédiatement à gauche et en dessous alternativement.

On peut résumer la méthode dans l'algorithme suivant :

1-  $i=1, j=m$

2-  $C_{ij} = \min(a_i ; b_j)$ . Si  $C_{ij} = a_i$  passé à (3) sinon passer à (4).

3- Poser  $b_j = b_j - a_i$  et  $i=i+1$ , si  $i \leq n$  passer à (2) sinon fin.

4- Poser  $a_i = a_i - b_j$  et  $j=j-1$ , si  $j \leq m$  passer à (2) sinon fin.

**Algorithme de NE :**

**Début :**

**Répéter !** (disponibilité = restriction = 0)

- $i, j \leftarrow$  aller vers Nord-Est disponible ;
- $C_{ij} \leftarrow \text{Min (Restriction } j ; \text{Disponibilité } i)$  ;
- $\text{Restriction } j \leftarrow \text{Restriction } j - C_{ij}$  ;
- $\text{Disponibilité } i \leftarrow \text{Disponibilité } i - C_{ij}$  ;
- Si ( $\text{Restriction } j == 0$ ) supprimer colonne  $j$  ;
- Si ( $\text{Disponibilité } i == 0$ ) supprimer ligne  $i$  ;

**Fin Répéter**

- Retourne tableau ;
- Retourne fct obj {cout unitaire  $ij$  \* unité  $ij$  } ;

**Fin début**

**Algorithme 2.4.** pseudo-code de NE.

Pour bien comprendre l'idée de la méthode on va l'appliquer sur l'exemple précédent :

**NE**

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER	8	5	6	120
M'SILA	15	10	12	80
SETIF	3	9	10	80
Demande	150	70	60	280

Dans le coin Nord-Est on met la plus grande valeur qui va soit satisfaire une demande ou bien épuiser une disponibilité. Dans l'exemple on va épuiser la 1ère disponibilité.

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER			60 $\rightarrow$	120-60 = 60
M'SILA				80
SETIF				80
Demande	150	70	60-60=0	220

Ensuite on refait la même chose mais sur le nouveau sous tableau qu'on obtient après la 1ère modification.

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER		60 $\downarrow$	60	60-60 = 0
M'SILA				80
SETIF				80
Demande	150	70-60=10	0	160

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER		60	60	0
M'SILA		10 →		80-10=70
SETIF				80
Demande	150	10-10=0	0	150

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER		60	60	0
M'SILA	70 ↓	10		70-70=0
SETIF				80
Demande	150-70=80	0	0	80

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER		60	60	0
M'SILA	70	10		0
SETIF	80			80-80=0
Demande	80-80=0	0	0	0

A la fin on obtient notre solution de base qui vérifie les contraintes de problème, donc la solution de base ainsi obtenue est représenté dans le tableau suivant :

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER		60	60	120
M'SILA	70	10		80
SETIF	80			80
Demande	150	70	60	280

Le coût total de cette solution :

$$Z = (60*5) + (60*6) + (70*15) + (10*10) + (80*3) = 2050$$

### 2.1.5 Nouvelle Heuristique : Sud-Ouest (SO) :

Cette méthode aussi est inspirée de la méthode nord-ouest. Mais on démarre du coin sud-ouest en utilisant la même mécanique.

Principe :

On considère à chaque étape, la case la plus Sud à l'Ouest de la matrice des coûts. On part donc de la route (i n ; j 1) ; on sature soit la ligne i n soit la colonne j 1. Puis on recommence sur la sous-grille formée des lignes et des colonnes non saturées.

Donc l'idée de la méthode est de remplir au maximum la case du tableau en haut, à gauche, puis compléter sur la ligne ou la colonne (de façon à atteindre l'offre ou la demande) et continuer ainsi à compléter les cases immédiatement à droite et en dessous alternativement.

On peut résumer la méthode dans l'algorithme suivant :

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

**Algorithme de SO:**

**Début :**

**Répéter !** (disponibilité = restriction = 0)

- $i, j \leftarrow$  aller vers Sud-Ouest disponible;
- $C_{ij} \leftarrow \min(\text{Restriction } j; \text{Disponibilité } i)$ ;
- $\text{Restriction } j \leftarrow \text{Restriction } j - C_{ij}$  ;
- $\text{Disponibilité } i \leftarrow \text{Disponibilité } i - C_{ij}$  ;
- Si ( $\text{Restriction } j == 0$ ) supprimer colonne  $j$  ;
- Si ( $\text{Disponibilité } i == 0$ ) supprimer ligne  $i$  ;

**Fin Répéter**

- Retourne tableau ;
- Retourne fct obj {cout unitaire  $ij$  \* unité  $ij$  } ;

**Fin début**

**Algorithme 2.5.** pseudo-code de SO

Pour bien comprendre l'idée de la méthode on va l'appliquer sur l'exemple précédent :

<b>SO</b>	ORAN	ANNABA	BISKRA	Disponibilité
ALGER	8	5	6	120
M'SILA	15	10	12	80
SETIF	3	9	10	80
Demande	150	70	60	280

Dans le Sud-Ouest on met la plus grande valeur qui va soit satisfaire une demande ou bien épuiser une disponibilité. Dans l'exemple on va épuiser la 1ère disponibilité.

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER				120
M'SILA				80
SETIF	80			80-80=0
Demande	150-80=70	70	60	200

Ensuite en refait la même chose mais sur le nouveau sous tableau qu'on obtient après la 1ere modification.

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER				120
M'SILA	70 →			80-70=10
SETIF	80			0
Demande	70-70=0	70	60	130

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER				120
M'SILA	70	10 ↓		10-10 = 0
SETIF	80			0
Demande	0	70-10 = 60	60	120

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER		60 →		120 - 60 = 60
M'SILA	70	10		0
SETIF	80			0
Demande	0	60-60=0	60	60

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER		60	60	60-60=0
M'SILA	70	10		0
SETIF	80			0
Demande	0	0	60-60=0	0

A la fin on obtient notre solution de base qui vérifie les contraintes de problème, donc la solution de base ainsi obtenue est représenté dans le tableau suivant :

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER		60	60	120
M'SILA	70	10		80
SETIF	80			80
Demande	150	70	60	280

Le coût total de cette solution :

$$Z = (60 \times 5) + (60 \times 6) + (70 \times 15) + (10 \times 10) + (80 \times 3) = 2050$$

### 2.1.6 Nouvelle Heuristique : Sud-Est (SE) :

Cette méthode est aussi inspirée de la méthode nord-ouest. Mais en démarrant du Coin sud-est en suivant le même processus.

Principe :

On considère à chaque étape, la case la plus Sud à l'Est de la matrice des coûts. On part donc

de la route ( $i \rightarrow j$ ) ; on sature soit la ligne  $i$  soit la colonne  $j$ . Puis on recommence sur la sous-grille formée des lignes et des colonnes non saturées. Donc L'idée de la méthode est de remplir au maximum la case du tableau en haut, à droite, puis compléter sur la ligne ou la colonne (de façon à atteindre l'offre ou la demande) et continuer ainsi à compléter les cases immédiatement à gauche et en dessous alternativement.

On peut résumer la méthode dans l'algorithme suivant :

- 1-  $i = n, j = m$
- 2-  $C_{ij} = \min(a_i ; b_j)$ . Si  $C_{ij} = a_i$  passé à (3) sinon passer à (4).
- 3- Poser  $b_j = b_j - a_i$  et  $i = i - 1$ , si  $i \leq n$  passer à (2) sinon fin.
- 4- Poser  $a_i = a_i - b_j$  et  $j = j - 1$ , si  $j \leq m$  passer à (2) sinon fin.

**Algorithme de SE:**

**Début :**

**Répéter !** (disponibilité = restriction = 0)

- $i, j \leftarrow$  aller vers Sud-Est disponible;
- $C_{ij} \leftarrow \min(\text{Restriction } j ; \text{Disponibilité } i)$ ;
- $\text{Restriction } j \leftarrow \text{Restriction } j - C_{ij}$  ;
- $\text{Disponibilité } i \leftarrow \text{Disponibilité } i - C_{ij}$  ;
- Si ( $\text{Restriction } j == 0$ ) supprimer colonne  $j$  ;
- Si ( $\text{Disponibilité } i == 0$ ) supprimer ligne  $i$  ;

**Fin Répéter**

Retourne tableau ;

Retourne fct obj {cout unitaire  $ij$  \* unité  $ij$  } ;

**Fin début**

**Algorithme 2.6.** pseudo-code de SE.

Pour bien comprendre l'idée de la méthode on va l'appliquer sur l'exemple précédent :

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER	8	5	6	120
M'SILA	15	10	12	80
SETIF	3	9	10	80
Demande	150	70	60	280

SE

Dans le Sud-Est on met la plus grande valeur qui va soit satisfaire une demande ou bien épuiser une disponibilité. Dans l'exemple on va épuiser la 1ère disponibilité

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER				120
M'SILA				80
SETIF			60 →	80-60=20
Demande	150	70	60-60=0	220

Ensuite on refait la même chose mais sur le nouveau sous tableau qu'on obtient après la 1ère modification.

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER				120
M'SILA				80
SETIF		20 ↓	60	20-20=0
Demande	150	70-20=50	0	200

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER				120
M'SILA		50 →		80-50=30
SETIF		20	60	0
Demande	150	50-50=0	0	150

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER				120
M'SILA	30 ↓	50		30-30=0
SETIF		20	60	0
Demande	150-30=120	0	0	120

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER	120			120-120=0
M'SILA	30	50		0
SETIF		20	60	0
Demande	120-120=0	0	0	0

A la fin on obtient notre solution de base qui vérifie les contraintes de problème, donc la solution de base ainsi obtenue est représentée dans le tableau suivant :

	ORAN	ANNABA	BISKRA	Disponibilité
ALGER	120			120
M'SILA	30	50		80
SETIF		20	60	80
Demande	150	70	60	280

Le coût total de cette solution :

$$Z = (60 \times 10) + (20 \times 9) + (50 \times 10) + (30 \times 15) + (120 \times 8) = 2690$$



Nous avons remarqué après avoir effectué plusieurs opérations que

NO=SE et SO=NE alors et en fin Nous en concluons que nous avons deux méthodes approchées pour résoudre le problème de transport la méthode de Coin Nord-Ouest et la nouvelle Heuristique la méthode Sud-Ouest. Alors on définit la méthode

Min-Ouest= Min (NO, SO)

### 2.1.7 Nouvelle Heuristique : MIN-Ouest (NO; SO) :

Ici on implémente les deux méthodes : coin nord-ouest (NO) et coin sud-ouest (SO) et on choisit la valeur minimal trouvée des deux méthodes. Pour bien comprendre l'idée de la méthode on va l'appliquer sur l'exemple précédant :

Méthode	Cout total
NO	2690
SO	2050
Min-Ouest (NO ; SO)	2050

## 3. Langage de la programmation et des algorithmes :

### 3.1 Définition du langage Java :

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au Sun World. La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java. La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java. Le langage Java reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, ou l'héritage multiple contourné par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.).[17]

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et

GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de greffons. Il offre toutes les facilités d'un IDE moderne (éditeur en couleurs, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web Compilé 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.[18]

#### 4. Simulation numérique :

Pour pouvoir juger la qualité de performance de notre algorithme proposé, nous avons généré des différents data dans la loi uniforme  $c_i \in [c_i, c_i']$ . Où pour chaque  $i$  ; 5 problèmes ont été créés induisant au total 45 problèmes. Le langage Java a été utilisé pour la programmation des algorithmes, qui ont été exécutés sur notre PC : NetBeans IDE 8.0.2, Windows 7, Système d'exploitation 64 bits, processeur : Intel(R) Core (TM) i3-2348 M CPU @ 2.30GHz 2.30 GHz, RAM : 4Go, La Marque HP.

##### 4.1 Génération des données :

Pour pouvoir comparer l'efficacité des méthodes considérées d'une façon plus objective différents problèmes ont été générés du terme suivant :

Après aussi générer les  $a_i$  et  $b_j$  origines et destinations de telle arc que  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$

les sont contrainte sont générés aléatoirement en sur les 9 intervalles des lois uniformes

[1-9], [10-99], [100-999], [1000-9999], [10000-99999], [100000-999999], [1000000-9999999], [10000000-99999999], [100000000-999999999], on pour chaque intervalle 5

problèmes ont été exécutés. Voir le tableau ci-dessous qui en résume pour chaque 5 problèmes la valeur du coût moyen. En fait pour traiter 45 problèmes ont été obtenus des résultats en utilisant les 5 méthodes.

##### 4.2 Discussion des résultats numériques :

Remarque : Bien sûr il est tout à fait clair que Min-Ouest(MO) domine Nord-Ouest(NO) et Sud-Ouest(SO) dans tous les cas et ceci par construction de Min-Ouest. Malgré ça la méthode de Vogel reste la dominante pour  $C_i \leq 99999999$ , Sur tous les méthodes. Alors

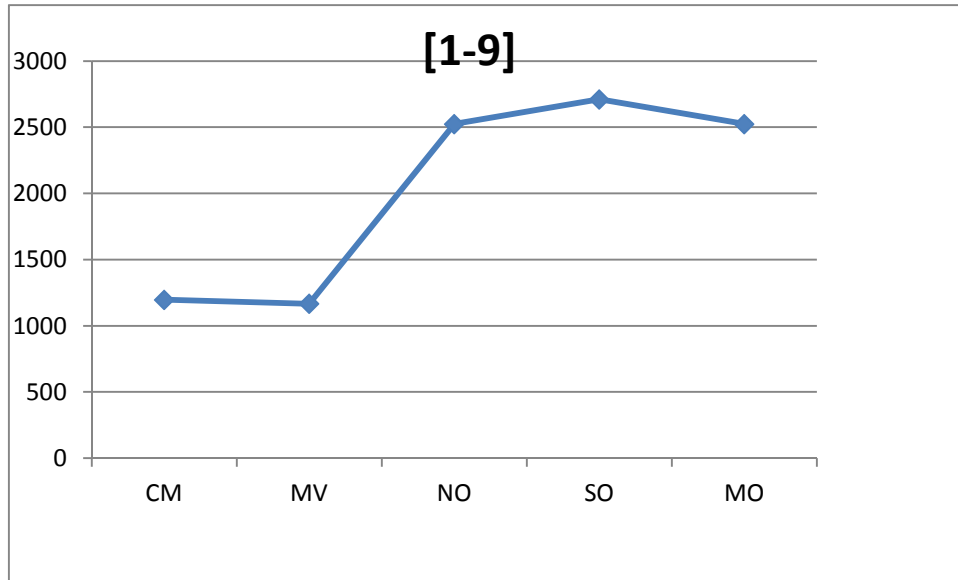
que pour  $C_i$  100000000,  $C_i \leq 999999999$  Min méthode de Vogel domine tous les autres méthodes. Cependant notre méthode reste efficace car elle est facile à manipuler et implémenter elle domine a au moins la méthode connu Nord-Ouest et ceci dans tous les cas.

<b>CI</b> <b>M</b>	[1-9]	[10-99]	[100-999]	[1000-9999]	[10000-99999]	[100000-999999]	[1000000-9999999]	[10000000-99999999]	[100000000-999999999]
CM	1197	16503	1,7 <sup>E5</sup>	1,55 <sup>E6</sup>	1,64E7	1,72E8	1,52E9	1,70E10	1,76E11
VOG	1167	1567	1,6 <sup>E5</sup>	1,44 <sup>E6</sup>	1,53 <sup>E7</sup>	1,55E8	1,54E9	1,61E10	1,69E11
NO	2525	28137	3,1 <sup>E5</sup>	3,03 <sup>E6</sup>	2,84E7	3,15E8	2,82E9	3,41E10	3,16E11
SO	2711	29323	3,2 <sup>E5</sup>	3,24 <sup>E6</sup>	2,63E7	2,88E8	3,21E9	2,63E10	3,44E11
Min-Ouest (NO ; SO)	2525	28137	3,1 <sup>E5</sup>	3,03 <sup>E6</sup>	2,63E7	2,88E8	3,21E9	2,63E10	3,16E11

**Table2.2 :** cout unitaire pour chaque intervalle par méthode.

problème	CM	MV	NO	SO	MO
[1-9]	1197	1167	2525	2711	2525

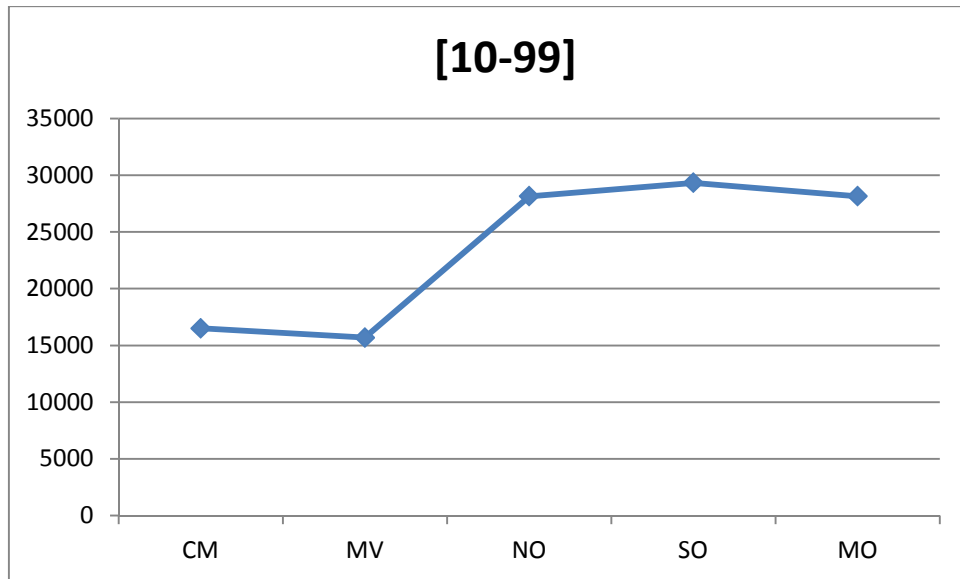
**Table2.3 :** cout unitaire pour chaque méthode par intervalle.



**Figure2.1:** diagramme de comparaison numérique les cinq méthodes par intervalle.

problème	CM	MV	NO	SO	MO
[10-99]	16503	15679	28137	29323	28137

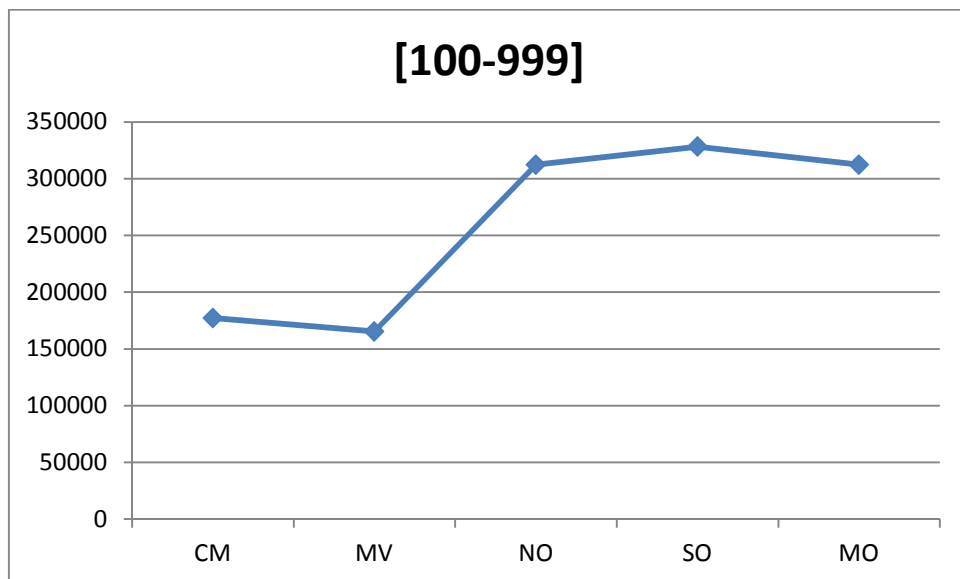
**Table2.4 :** cout unitaire pour chaque méthode par intervalle.



**Figure 2.2:** diagramme de comparaison numérique les cinq méthodes par intervalle.

problème	CM	MV	NO	SO	MO
[100-999]	177143	165268	312373	328211	312373

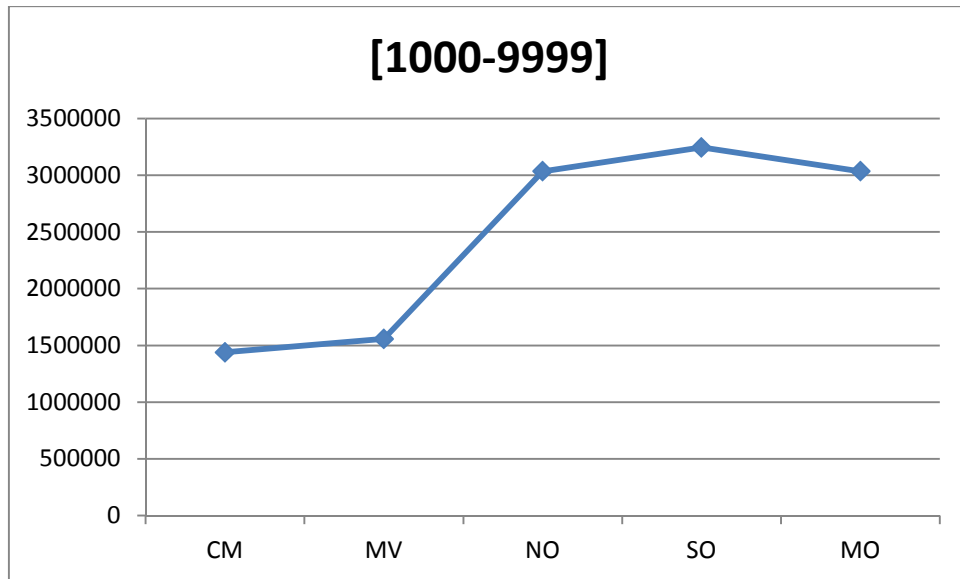
**Table2.5:** cout unitaire pour chaque méthode par intervalle.



**Figure 2.3:** diagramme de comparaison numérique les cinq méthodes par intervalle.

problème	CM	MV	NO	SO	MO
[1000-9999]	1439367	1557673	3033862	3243728	3033862

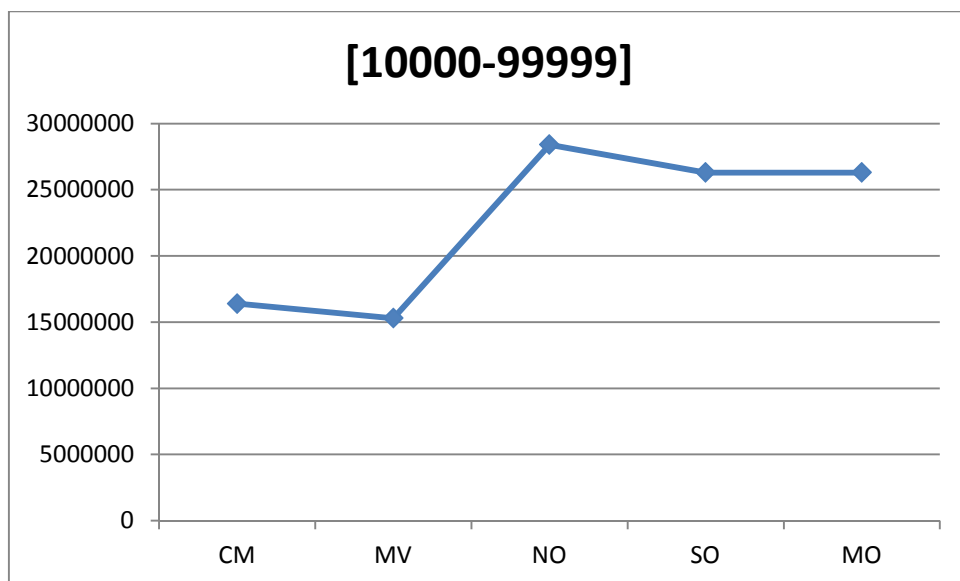
**Table2.6:** cout unitaire pour chaque méthode par intervalle.



**Figure 2 .4:** diagramme de comparaison numérique les cinq méthodes par intervalle.

problème	CM	MV	NO	SO	MO
[10000-99999]	16400000	15300000	28400000	26300000	26300000

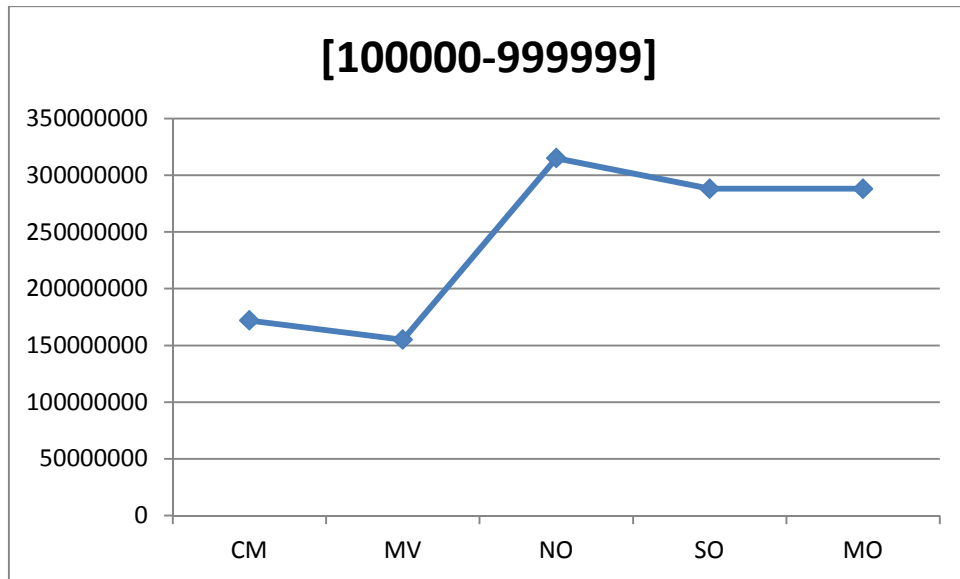
**Table.2.7:** cout unitaire pour chaque méthode par intervalle.



**Figure 2 .5:** diagramme de comparaison numérique les cinq méthodes par intervalle.

problème	CM	MV	NO	SO	MO
[100000-999999]	172000000	155000000	315000000	288000000	288000000

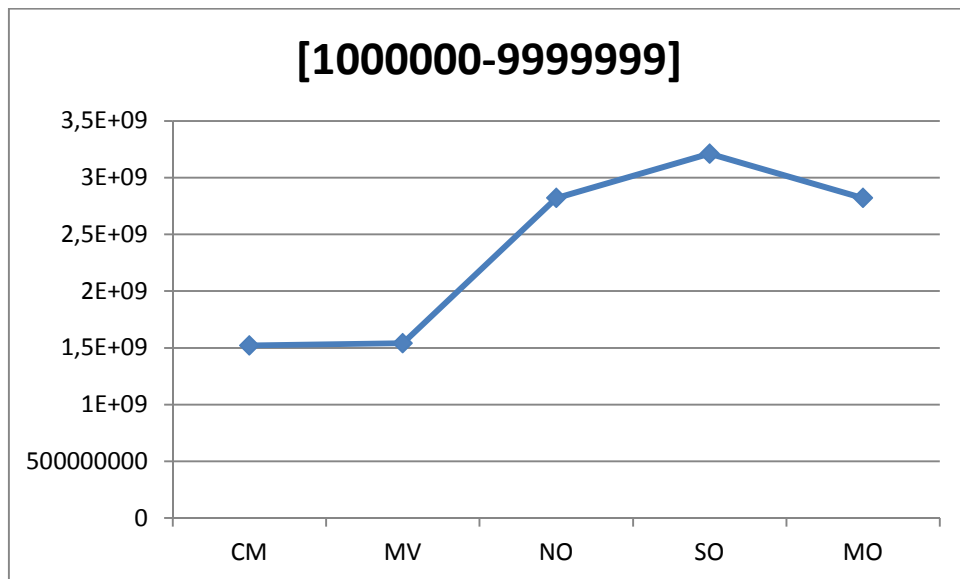
**Table2.8:** cout unitaire pour chaque méthode par intervalle.



**Figure 2.6:** diagramme de comparaison numérique les cinq méthodes par intervalle.

problème	CM	MV	NO	SO	MO
[1000000-9999999]	1520000000	1540000000	2820000000	3210000000	2820000000

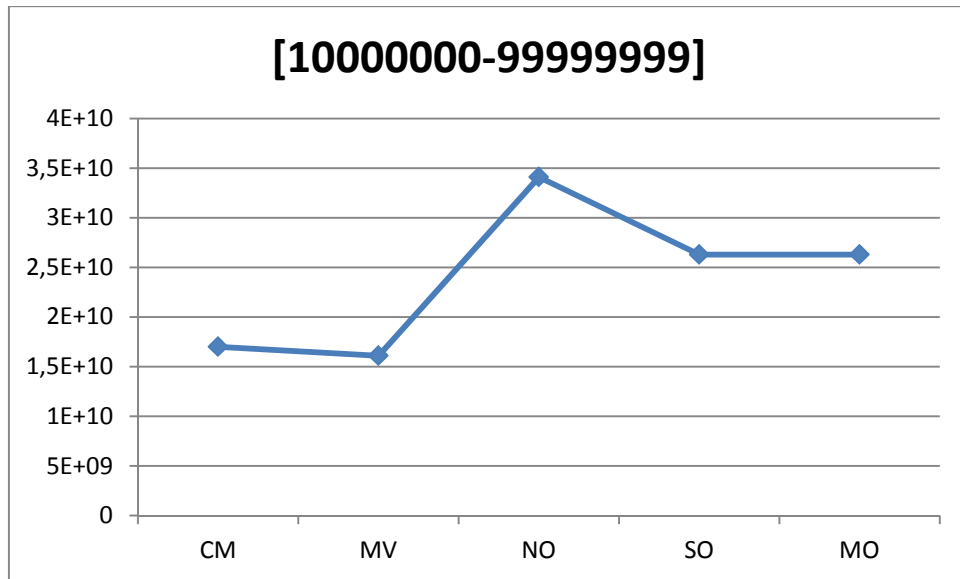
**Table2.9:** cout unitaire pour chaque méthode par intervalle.



**Figure 2.7:** diagramme de comparaison numérique les cinq méthodes par intervalle.

problème	CM	MV	NO	SO	MO
[10000000-99999999]	17000000000	16100000000	34100000000	26300000000	26300000000

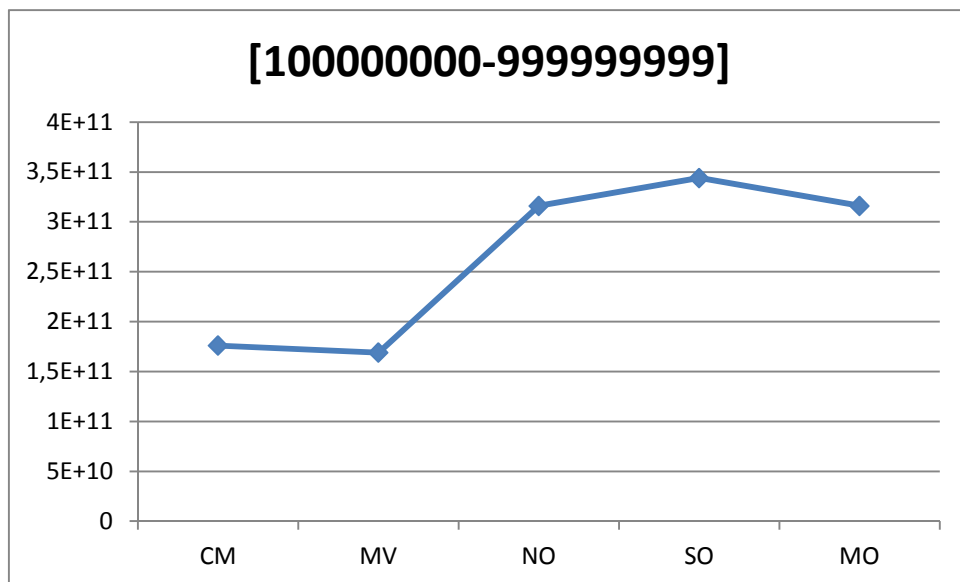
**Table2.10:** cout unitaire pour chaque méthode par intervalle.



**Figure 2.8:** diagramme de comparaison numérique les cinq méthodes par intervalle

problème	CM	MV	NO	SO	MO
[100000000-999999999]	1,76E+11	1,69E+11	3,16E+11	3,44E+11	3,16E+11

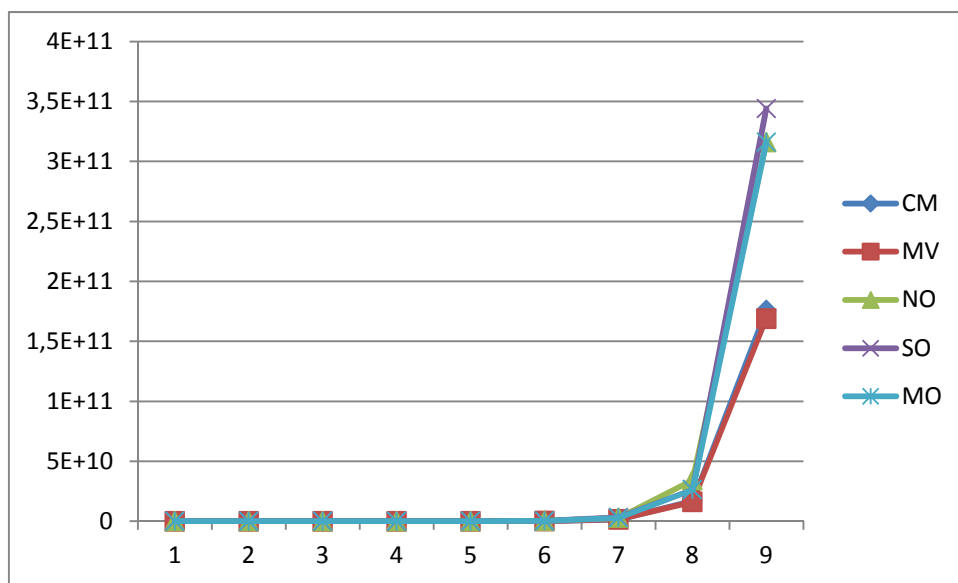
**Table2.11:** cout unitaire pour chaque méthode par intervalle.



**Figure 2.9:** diagramme de comparaison numérique les cinq méthodes par intervalle.

$C_i \backslash M$	CM	MV	NO	SO	MO
1	1197	1167	2525	2711	2525
2	16503	15679	28137	29323	28137
3	177143	165268	312373	328211	312373
4	1439367	1557673	3033862	3243728	3033862
5	16400000	15300000	28400000	26300000	26300000
6	172000000	155000000	315000000	288000000	288000000
7	1520000000	1540000000	2820000000	3210000000	2820000000
8	1,7E+10	16100000000	3,41E+10	2,63E+10	26300000000
9	1,76E+11	1,69E+11	3,16E+11	3,44E+11	3,16E+11

**Table2.12:** cout unitaire pour chaque méthode pour tous les intervalles.



**Figure 2.10:** diagramme générale de comparaison numérique les cinq méthodes pour tous les intervalles.

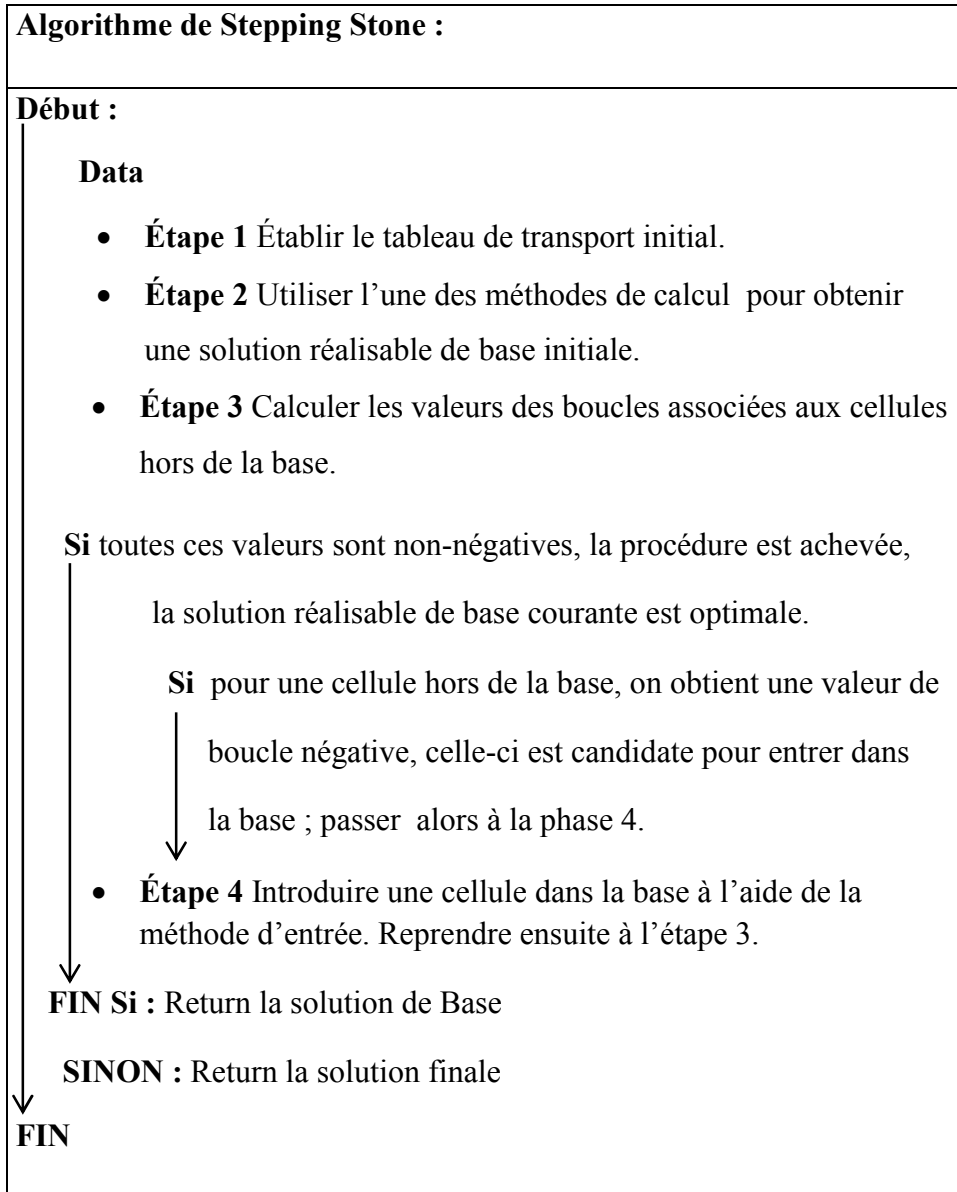
## 5. Méthode de solution exacte (Stepping-Stone) :

La méthode Stepping-Stone permet de vérifier l'optimalité de la solution possible initiale déterminée en utilisant l'une des méthodes. Le coin Nord-Ouest(NO), la méthode à moindre coût(CM), la méthode d'approximation de Vogel(MV), la méthode Sud-Ouest(SO), ou la méthode Min-Ouest(MO). Ainsi, la méthode de la pierre intermédiaire est une procédure pour trouver le potentiel de toute variable non-basique (cellules vides) en fonction de la fonction objective. Grâce à la méthode Stepping-Stone, nous déterminons que l'effet sur le coût de transport serait au cas où une unité serait affectée à la cellule vide. À l'aide de cette méthode, nous découvrons si la solution est optimale ou non. La série d'étapes consiste à vérifier l'optimalité de la solution réalisable initiale à l'aide de la méthode de la pierre intermédiaire :



Principe :

1. La condition préalable à la résolution de l'optimalité est de s'assurer que le nombre de cellules occupées est exactement égal à  $m + n - 1$ , où 'm' est le nombre de lignes, tandis que 'n' est égal au nombre de colonnes.
2. Tout d'abord, la cellule vide est sélectionnée, puis le chemin fermé est créé qui commence à partir de la cellule inoccupée et retourne dans la même cellule inoccupée appelée "boucle fermée". Pour créer une boucle fermée, il faut tenir compte des conditions suivantes:
  - En boucle fermée, les cellules sont sélectionnées dans une séquence telle qu'une cellule est inutilisée / inoccupée et toutes les autres cellules sont utilisées / occupées.
  - Une paire de cellules consécutives utilisées se trouve soit dans la même rangée, soit dans la même colonne.
  - Aucune cellule occupée consécutive ne peut être dans la même ligne ou colonne.
  - La première et la dernière cellule en boucle fermée se trouvent soit dans la même rangée ou dans la même colonne.
  - Seuls les mouvements horizontaux et verticaux sont autorisés.
3. Une fois que la boucle est créée, attribuez un signe "+" ou "-" alternativement sur chaque cellule d'angle de la boucle, mais commencez avec le signe "+" pour la cellule inoccupée.
4. Répétez ces étapes jusqu'à ce que toutes les cellules inoccupées soient évaluées.
5. Maintenant, si tous les changements calculés sont positifs ou sont égaux ou supérieurs à zéro, la solution optimale a été atteinte.
6. Mais dans le cas, le cas échéant, la valeur est négative, il est possible de réduire davantage le coût du transport. Ensuite, sélectionnez cette cellule inoccupée qui a le changement le plus négatif et attribuez autant d'unités que possible. Soustrayez l'unité qui a ajouté à la cellule inoccupée des autres cellules avec un signe négatif en boucle, afin d'équilibrer les exigences de demande et d'approvisionnement.



**Algorithm 2.7** Pseudo code de Stepping-Stone.

## 6. Conclusion :

L'importance d'une solution initiale meilleure joue un grand rôle dans l'efficacité de la méthode exacte utilisant celle-ci. Ceci nous a incités de développer une méthode dont son application au problème de transport améliore la solution initiale trouvée par ce qu'on appelle la méthode du Coin Nord-Ouest(NO). C'est en fait une méthode qu'on a inspiré de celle-ci. C'est en fait elle est de duite de celle-ci et celle du la méthode Coin Sud-Ouest(SO).