Ministre de l'enseignement supérieur et de la recherche scientifique
Université Abderrahmane Mira Béjaïa
Faculté des sciences exactes
Département de recherche opérationnelle

Mémoire de Fin de cycle

En vu d'obtention du Diplôme de Master en mathématique appliqué

Option : Modélisation Mathématique et technique de décision

Thème

Modélisation et Résolution du Probléme de Transport (Marchendise)

Réalisé par :

✓ **Kebbi Souad**

✓ **Messaoud said**

Encadré par :

✓ **Dr kabyl.K**

Devant le jury

Président	Mr TAOUINET Smail	U. A/Mira Béjaia.
Examinatrice	M YOUNSI Leila	U. A/Mira Béjaia

Promotion 2019-2020

"Les thèses les plus fausses sont souvent les plus belles."

Pierre Daninos

Remerciements

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

Nous tenons tout d'abord à remercier ***Dr Kabyl.K****, 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 exprimons notre grand respect aux honorables membres de jury qui ont accepté d'évaluer ce 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 famille et nos amis pour le soutien qu'ils nous ont apportés durant toutes ces années d'études et tous ceux qui nous ont soutenus de prés ou de loin durant tout notre cursus et espérons que ce mémoire servira de guide pour les promotions à avenir.

Dédicaces

A cœur veillant, rien d'impossible. A conscience tranquille, tout est accessible. Quand il y a la soif d'apprendre. Tout vient à point à qui sait attendre. Les études sont avant tout notre unique et seul atout. Souhaitant que le fruit de nos efforts fournis jour et nuit Nous mènera vers le bonheur fleuri.

Je dédie ce modeste travail :

A la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur. et ma réussite, à ***ma mère*** que j'adore.

A ***mon père***, l'homme qui m'a donné le désir d'apprendre et le savoir vivre, école de mon enfance, et qui a tant attendu ce moment avec impatience, qui a été mon ombre durant toutes les années des études, Que dieu les gardes et les protégé.

A mes très chéres soeurs {Zahra, Lyticia, lina}.

A mon très chére cousin {Massine}.

A toute ma famille.

A mon binôme Said

A tous mes amis avec lesquels j'ai partagé mes moments de joie et de bonheur {Kamilia, Feyrouz, Sarah, Nassima, lyes, yanis, Mounir, Hamza. }

Kebbi Souad

Dédicaces

Je dédie ce modeste travail :

A ma **très chère mère** Quoi que je fasse ou que je dise,je ne suerai point te remercier comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta présence a mes cotes a toujours été ma source de force pour affronter les différents obstacles.

A **mon cher père**, à mes grands-parents qui ont toujours était là pour moi.

A (Amirouche, Samir, Zahira, Saida, Zico, Kahina) que je considère comme mes frères et sœurs vous avez toujours été la à mes cotés pour me soutenir et m'encourager. Que se travail traduit ma gratitude et mon affection.

A mes tentes que j'aime énormément (Fatiha, Ounissa, Saliha, Djamila, Akila).

A mes cousin (farouk, Redouan, Sofiane, Malikou, Riad, Yahia, Lounis, Ghilas).

A mes cousines (khadidja ,Nesrine ,Lahna ,Damia ,Dihia ,Zinouba ,Fatma ,Cilia , Amina ,Yamina ,Sabrina ,Taous ,Thiziri ,Massilva ,Cilia ,Yefsan ,Lahna)

A ma Binome Souad.

A mes amis proche (Djarri, Yacine, khelaf, Nabil, Mourad, Aziz , Mourad Imoula, Younes, Bachir, khlifa, karim, Bakli, kiki).

A tous ceux que j'aime.

Merci

Messouad Said

TABLE DES MATIÈRES

table des Figures v

Introduction **1**

1 QUELQUES NOTIONS ET NOTATIONS DE BASE SUR LA THÉORIE DES GRAPHES **3**

1.1 Généralités sur les graphes 3

1.1.1 Qu'est ce qu'un graphe 3

1.1.2 Graphe simple, Graphe Multiple, Graphe Complémentaire 5

1.1.3 graphe orienté et non orienté, graphe valué 5

1.1.4 Sous-graphe, Graphe partiel,Sous-graphe partiel 7

1.1.5 Source, puits 8

1.1.6 Clique 8

1.1.7 Stable 8

1.2 Chaîne, Chemin, Cycle, Cocycles et Circuit 9

1.2.1 Chaîne 9

1.2.2 Chemin 9

1.2.3 Cycle Élémentaire 10

1.2.4 Cocycles 10

1.2.5 Circuit 10

1.3 Connexité 10

1.3.1 Graphe Connexe 10

1.3.2 Connexité dans les graphes : 11

1.3.3 Flux et Flots 13

1.3.4 Chaîne améliorante 13

1.4 Représentations matricielle des graphes 14

1.4.1 Matrice d'adjacence . . . 14
1.4.2 Matrice d'incidence . . . 14
1.5 Quelques graphes particuliers . . . 15
1.5.1 Graphe complet : . . . 15
1.5.2 Graphe d'écart : . . . 16
1.5.3 Graphe biparti : . . . 16
1.5.4 Graphe planaire : . . . 17
1.5.5 Arbre : . . . 18

2 Problème de Transport et Méthodes de Résolution 19
2.1 Programmation linéaire . . . 19
2.1.1 Forme générale d'un programme linéaire . . . 20
2.1.2 Formes matricielles classiques et conventions . . . 20
2.2 La complexité algorithmique . . . 21
2.2.1 La notation O(.) . . . 21
2.2.2 Calcule de la complixité d'un algorithme . . . 22
2.3 Positionnement du problème . . . 23
2.4 Modélisation . . . 23
2.4.1 Variables de décision . . . 23
2.4.2 Fonction objective . . . 24
2.4.3 Contraintes . . . 24
2.4.4 Formulation mathématique . . . 24
2.5 Tableau de transport . . . 25
2.6 Réseau de transport . . . 26
2.7 Dégénérescence en problème de transport : . . . 27
2.8 Structure de la résolution de problème de transport : . . . 27
2.8.1 Solution de base réalisable . . . 28
2.8.2 Solution optimale . . . 28
2.8.3 Organigramme de résolution pour le problème de transport . . . 30
2.8.4 Algorithme général de résolution de problème de transport . . . 31
2.9 Méthode de détermination de la solution de base . . . 32
2.9.1 Méthode du COIN NORD-OUEST . . . 32
2.9.2 Méthode de coût minimum . . . 35
2.9.3 Méthode des pénalité (Balas-hammer) . . . 42
2.10 Méthode d'optimisation de la solution de base . . . 46
2.10.1 Méthode de Stepping-Stone . . . 46
2.10.2 Méthode de Distribution Modifiée . . . 49
2.11 Problème d'affectation . . . 51

2.11.1 Résolution du problème d'affectation 52
2.12 Résolution du Problème de flot 54
2.12.1 Algorithme de Ford Ful kerson 55

3 Résolution de quelque problème de transport 61
3.1 Résolution de problème de transport containeurs 61
3.1.1 Détermination d'une solution de base initiale : 61
3.1.2 Formulation mathématique 62
3.1.3 Optimisation de solution de base initiale : 70
3.1.4 Méthode de distribution modifiée : 76
3.2 Application de l'algorithme Hongrois 79
3.2.1 Problème 2 79
3.2.2 Formulation mathématique 81
3.3 Application de l'algorithme de Ford ful kerson 84
3.3.1 problème d'alimentation de 4 villes par 3 châteaux d'eau : 84

4 APPLICATION ET ÉVALUATION 91
4.1 Langages utilisés 91
4.2 Technologies utilisées 92
4.2.1 Fichier nouveau 93
4.2.2 Fichier existant 93
4.2.3 Fichier «compilé» et «exécuté» 93
4.3 Excution de la méthode coin Nord oust et cout minimum 97
4.3.1 la Méthode coin N O 97
4.3.2 Cout minimum 99
4.4 Résultat d'exécution de problème d'affectation 100
4.4.1 Complexité de l'algorithme 100
4.5 Résultat d'exécution de problème de folt 101
4.5.1 Complexité de l'algorithme 102

Conclusion 103

TABLE DES FIGURES

1.1 graphe à 5 sommets et 5 arcs 4
1.2 Graphe G et son Complémentaire 5
1.3 Graphe orienté 6
1.4 Graphe non orienté 7
1.5 Sous-graphe G 7
1.6 Graphe G et son graphe partiel 8
1.7 Un graphe a deux stable 9
1.8 Chemin 9
1.9 Un graphe non orienté connexe 11
1.10 Composantes connexes d'un graphe 11
1.11 Graphe fortement connexe 12
1.12 Graphe fortement connexe 13
1.13 graphe à 6 sommets et 7 arcs et la matrice adjacence associée 14
1.14 graphe à 6 sommets et 5 arêtes sans boucle et la matrice d'incidence associée . 15
1.15 Graphe biparti 16
1.16 Graphe planaire 17
1.17 Graphe planaire 18

2.1 Tableau de transport 26
2.2 Réseau de transport 27
2.3 Organigramme de la résolution de problème de transport. 29
2.4 Graphe d'affectation de la méthode du Vogel et Balas-Hammer 46
2.5 Problème de flot maximum 57
2.6 Le réseau après avoir défini le nouveau flot x1 58
2.7 Le réseau après avoir défini le nouveau flot x2 58
2.8 Le réseau après avoir défini le nouveau flot x3 59

2.9 Le réseau après avoir défini le nouveau flot x4 60
2.10 Le réseau après avoir défini le flot maximum et la coupe minimal 60

3.1 le graphe assossis au problème de transport initial 62
3.2 Le graphe associé a la Table 4.2 65
3.3 le graphe associé au problème 2 80
3.4 85
3.5 problème de flot initial 85
3.6 Le réseau après avoir défini le nouveau flot 86
3.7 Le réseau après avoir défini le nouveau flot 87
3.8 Le réseau après avoir défini le nouveau flot 88
3.9 Le réseau après avoir défini le nouveau flot maximum 89
3.10 coupe minimal 90

4.1 Interface de Dev-C++ 92
4.2 Interface de Dev-C++ 95
4.3 problème de flot initial 101

INTRODUCTION

Introduction générale

La recherche opérationnelle (RO) est la discipline des mathématiques appliquées qui traite l'optimalité des ressources dans l'industrie. Depuis une dizaine d'année, le champ d'application de la RO s'élargi à des domaine comme l'économie, la finance, le marketing et la planification d'entreprise. Plus récemment, la RO a été utilisée pour la gestion des système de santé et d'éducation, pour la résolution de problème environnementaux [9].

De nos jours, la recherche opérationnelle (RO) a été menée dans la plupart des cas dans le domaine civil, ses racines sont généralement attribuées au service militaire. En raison de son ampleur, la guerre mondiale a un besoin urgent d'une distribution efficace les ressources sont limitées à diverses opérations et activités militaires au sein de chaque chirurgie. Surtout les organisations militaires britanniques et américaines ont commencé de nombreux scientifiques contribuent à gérer ces distributions et à s'occuper d'autres questions stratégiques et tactiques.

Elle est née pendant la Seconde Guerre mondiale des mathématiciens exceptionnels (dont von Neumann, Danzg, Blackett) Personnes à qui l'on demande de fournir une technologie d'optimisation ressources militaires. Le premier succès de cette méthode est Patrick Blackett, lauréat du prix Nobel de physique en 1940, a résolu un problème optimiser l'installation du radar de surveillance. Qualificatifs "opérationnels "dés l'application initiale de la discipline Action militaire. Ce nom restera par la suite, même si le domaine le militaire n'est plus le principal champ d'application de la discipline quelle est la signification du terme «opérationnel» "efficace". c'est tout Mathématicien qui a créé une nouvelle méthode caractérisée par des mots-clés Modélisation et optimisation.

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 questions les plus fréquemment posées, nous avons répertorié les problèmes d'expédition, Le problème du transport est la programmation linéaire dans de nombreuses situations mérite l'attention surtout la recherche opérationnelle. C'est peut-être le problème La programmation linéaire spéciale la plus importante sur la fréquence relative Il apparaît dans l'application et aussi dans la simplicité du processus développé.

Ce manuscrit est organisé en cinq chapitres précédés par une introduction générale. Dans le premier chapitre nous définissons quelques éléments fondamentaux de la théorie des graphes suit d'un deuxième chapitre sur la complexité Algorithmique et la programmation linéaire. Le troisième chapitre est consacré à la détermination et la définition du problème de transport classique, affectation, et le problème du flot maximum ainsi que les algorithmes d'optimisation qui vont être utilisés pour la résolution de quelques problèmes . Dans le quatrième chapitre est consacré à la résolution de certains problèmes d'optimisation (problème de transport, affectation, flot maximum). Le cinquième chapitre est l'application de ces méthodes dans les réseaux en utilisant le langage C , les résultats obtenus, nous terminons le manuscrite par une conclusion générale et quelques perspectives.

CHAPITRE 1

QUELQUES NOTIONS ET NOTATIONS DE BASE SUR LA THÉORIE DES GRAPHES

L'objet de ce chapitre est de présenter brièvement, certaines définitions de bases ou concepts généraux sur les éléments de la théorie des graphes nécessaires à une bonne compréhension de la suite du mémoire.

1.1 Généralités sur les graphes

1.1.1 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 paires de sommets v_1 et v_2 par une arête (arc) e, les points $v_1$et v_2 appelés les extrémités.[6]
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 un arête, et si $x = y$ alors e définit une boucle. en d'autre terme une boucle est une arête reliant un sommet à lui même.[6]
Le graphe $H = (V, E')$ est un graphe partiel de G, si $E' \subseteq E$, Autrement dit on obtient H en enlevant une ou plusieurs arêtes au graphe G.[1]
Un sous graphe engendré par $A \subseteq V$ est un graphe dont les sommets sont ceux de A et les arêtes sont celles ayant les deux extrémités dans A. on le note G_A[7]

La Figure ci-dessous présente un graphe à 5 sommets et 5 arcs

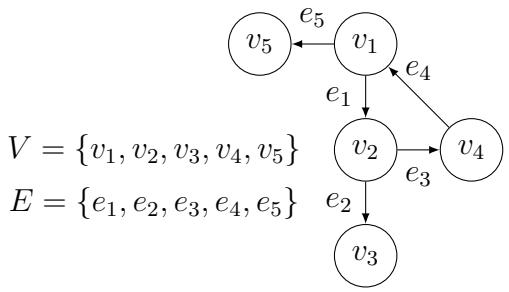


FIGURE 1.1 – graphe à 5 sommets et 5 arcs

Considérons le graphe correspondant à la Figure 1.1
- De v_1 on peut atteindre v_2 par e_1. Donc, v_1 forment l'ensemble des prédécesseurs de v_2, qu'on note $\Gamma^-(v_2)$.
- De v_2 on peut atteindre v_4 et v_3 par e_3 et e_2. Donc, v_4 et v_3 forment l'ensemble des successeurs de v_2, qu'on note Γ^+ (v_2).
- L'ensemble des voisins du sommet v_2 noté, $N(v_2)$ est égale à la réunion de l'ensemble de ses prédécesseurs et de ses successeurs.

Extrémité initiale et terminale (successeur et prédécesseur)

Soit un arc (i, j), j est dis extrémité terminale de (i, j).On dit aussi que j est successeur de i,et i un prédécesseur de j.L'arc (i, j) incident vers l'extérieur en i et vers l'intérieur en j[5].

Définition 1.1 un arc de G de la forme (i, i) est appelé une boucle. pour un arc $u = (i, j)$ le point i est son extrémité initial, et le point j son extrémité terminal.
on dis que j est un successeur de i s'il existe un arc ayant son extrémité initial en i et son extrémité terminal en j. L'ensemble des successeurs de i se note

$$\Gamma_G^+(i)$$

De même, on dit j est un prédécesseur de i s'il existe un arc de la forme (j, i). L'ensemble des prédécesseurs de i se note

$$\Gamma_G^-(i)$$

L'ensemble des sommets voisins de i se note

$$\Gamma_G(i) = \Gamma_G^+(i) \cup \Gamma_G^-(i)$$

On note $\Gamma_G(i)$ l'ensemble des successeurs de i, $\Gamma_G^-(i)$ L'ensemble des prédécesseurs de i, $d^+(i)$ le demi-degré extérieur de i, c'est-à-dire le cardinal de $\Gamma_G^+(i)$,et $\Gamma_G^-(i)$ le demi-degré intérieur de i,c'est-à-dire le cardinal de $\Gamma_G^-(i)$

1.1.2 Graphe simple, Graphe Multiple, Graphe Complémentaire

Graphe simple :

Un graphe G est dit simple si tous ses sommets sont sans boucles et entre chaque paire de sommets, il y a au plus une arête.[16]

Graphe Multiple

C'est un graphe qui possède des boucles ou bien des arêtes (arcs) multiples.[16]

Graphe Complémentaire :

Soit G= (X, E) un graphe simple. Le graphe $\bar{G}$=(v,$\bar{E}$) est le graphe complémentaire de G si :

$$\forall e \in E \Leftrightarrow e \notin \bar{E}$$

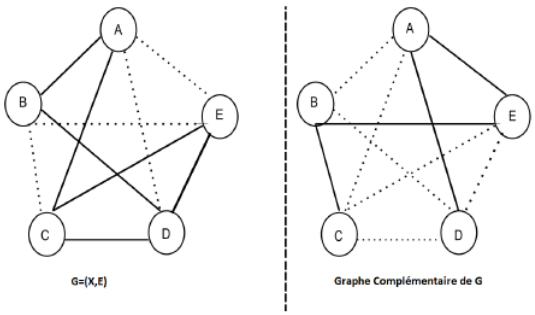


FIGURE 1.2 – Graphe G et son Complémentaire

1.1.3 graphe orienté et non orienté, graphe valué

Graphe orienté :

On dit que G est un graphe orienté (ou bien direct) s'il y a une distinction entre les liens (x ; y) et (y ; x), c'est-à-dire $(x; y) \neq (y; x)$. Dans ce cas le lien est appelé un arc. On représente

$\alpha = (x; y)$ graphiquement par une flèche qui part de x pour joindre y qui sera la pointe de cette flèche. Dans ce cas, y sera appelé un successeur de x (ou x est un prédécesseur de y) et chaque sommet peut avoir plusieurs successeurs et plusieurs prédécesseurs. On appelle une boucle tout arc (x ; x) dont ses extrémités se coïncident.
Le graphe de la figure 1.3 possède quatre sommets et quatre arcs qui sont :

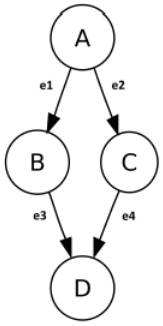


FIGURE 1.3 – Graphe orienté

X = $\{A, B, C, D\}$, $E = \{e_1, e_2, e_3, e_4\}$
Ce graphe est d'ordre n = 4 et de taille m = 4.

Graphe non orienté :

On dit que G est un graphe non orienté (ou bien indirect), si la précision de sens de lien (x ; y) et la distinction entre extrémité initiale et extrémité terminale ne jouent aucun rôle. On appelle tout élément $(x; y) \in E$ une arête, qui est représentée graphiquement par un segment sans flèche liant les deux nœuds x et y.[10]

Graphe valué

Un graphe valué $G = (X; U; v)$ est un graphe $(X; U)$ (orienté ou non orienté) muni d'une application $v : U \to \Re$. L'application v est appelée valuation du graphe. On peut étendre cette valuation en posant $\forall (x; y) \in X^2, v(x; y) = +\infty si (x; y) \notin U$ [14]

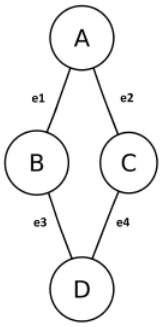


FIGURE 1.4 – Graphe non orienté

1.1.4 Sous-graphe, Graphe partiel,Sous-graphe partiel

Sous-graphe

Soit G = (X, E) un graphe. Soit A un sous ensemble de sommets inclus dans X et le sous graphe $G_A = (X_A, \text{E(A)})$ dont l'ensemble des sommets est A et d'arêtes est E(A) qu'est formé de toutes les arêtes de G ayant les deux extrémités dans A. Soit le graphe G suivant :

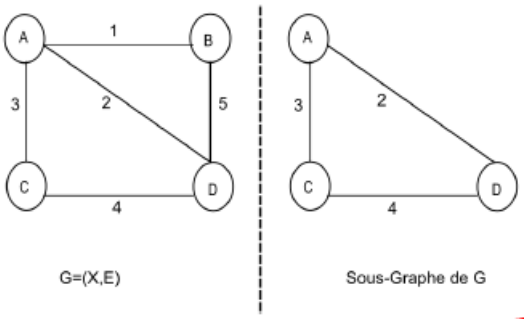


FIGURE 1.5 – Sous-graphe G

Graphe partiel

Soit G = (X, E) un graphe. Le graphe $G' = (X, E')$ est un graphe partiel de G si E' est inclus dans E. Autrement dit, on obtient G' en enlevant une ou plusieurs arête au graphe G.

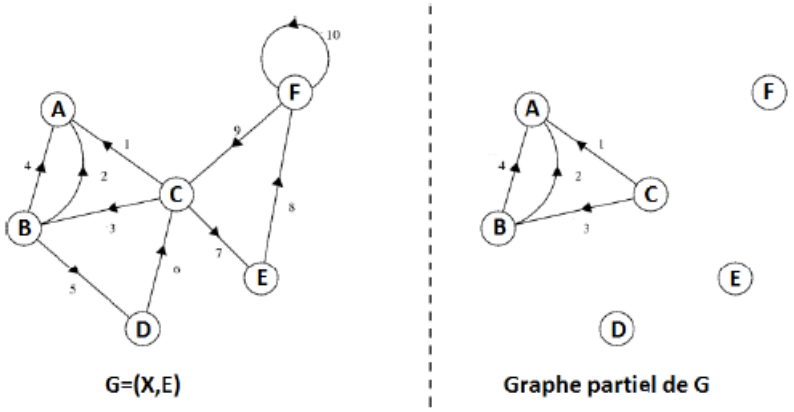


FIGURE 1.6 – Graphe G et son graphe partiel

Sous-graphe partiel

Un sous-graphe partiel est un graphe partiel d'un sous graphe.

1.1.5 Source, puits

une source S est un sommet ascendant de touts les autres sommets,un **puits** p, un sommet descendant de tous les autres sommets.

1.1.6 Clique

Soit $G = (V, E)$ tel que X un ensemble de sommets de G deux à deux adjacents.[18] Une clique de n sommets est noté K_n. Donc un graphe G est une clique si $d_G(x)$= n-1 pour tout ses sommets telle que $|X| = n$.

1.1.7 Stable

Un stable est un ensemble de sommets de G deux à deux non-adjacents. Le graphe de la Figure 1.7 admet deux stables tel que : $S1 = fA, C, H, Fg$ et $S2 = fG, D, B, Eg$

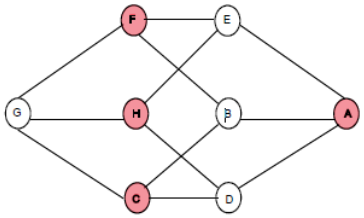


FIGURE 1.7 – Un graphe a deux stable

1.2 Chaîne, Chemin, Cycle, Cocycles et Circuit

1.2.1 Chaîne

Une chaîne dans un graphe est une suite $v_0, v_1, ..., v_k$ tel que v_i, $\forall i \in 1, 2, ..., k$, est reliés par une arête à v_{i-1} et par une autre arête à v_{i+1}.

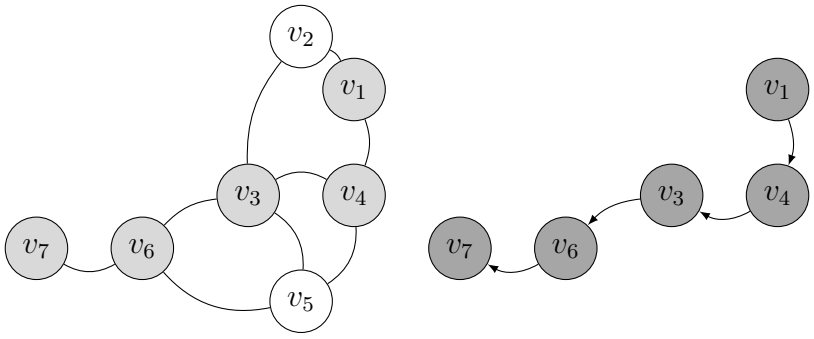
• La longueur d'une chaîne correspond au nombre d'arêtes parcourues

1.2.2 Chemin

Un **chemin** dans un graphe orienté est un suite de sommets $\{v_0, v_1, ..., v_k\}$ tel que $\forall i \in \{1, 2, ..., k\}$, $\exists$ un arcs de v_{i-1} vers v_i et un autre arc de v_i vers v_{i+1}

• Un chemin p est **simple** si chaque arc du chemin est empruntée une seule fois .

Le graphe de la figure 1.8(b) présente un chemin de longueur 4 dans le graphe de la figure 1.8(a)



(a) **(b)**

FIGURE 1.8 – Chemin

1.2.3 Cycle Élémentaire

Un **cycle** est une chaîne simple finissant à son point de départ.*i.e* on ne rencontre pas deux fois le même sommet, sauf celui choisi comme sommet de départ et d'arrivée.[19]

1.2.4 Cocycles

Soit $G = (V, E)$ un graphe et $Y \subset V, Y \neq \emptyset$. Considérons les sous ensembles d'arcs suivants :
• $w^+(Y)$ l'ensemble des arcs ayant leur extrémité initial dans Y et leur extrémité terminal dans X/Y c'est-a-dire :
$w^+(Y) = \{u_k \in U/I(u_k) \in Y \text{ et } T(u_k) \notin Y\}$.

• $w^-(Y)$ l'ensemble des arcs ayant leur extrémité terminal dans Y et leur extrémité initial dans X/Y c'est-a-dire :
$w^-(Y) = \{u_k \in U/I(u_k) \notin Y \text{ et } T(u_k) \in Y\}$.

l'ensemble $W(Y) = w^+(Y) \cup w^-(Y)$ est appelé cocycle relatif à Y.

1.2.5 Circuit

Dans un graphe orienté un **circuit** est un chemin tel que le sommet de départ est le même que celui d'arrivé, en d'autre terme c'est un chemin qui se ferme sur lui-même.[17]

1.3 Connexité

Un graphe connexe est un graphe tel que pour toute paire de sommets x et y, il existe une chaîne reliant x et y, un graphe non connexe est décomposé en plusieurs composantes connexes.

1.3.1 Graphe Connexe

Un graphe G est dit **connexe** si et seulement s'il existe une chained entre chaque paire de sommets, si G n'est pas connexe alors il est décomposé en plusieurs sous-graphes appelés **composantes connexes**.

1.3.2 Connexité dans les graphes :

Cas des graphes non orientés :

Un graphe non orienté est connexe si, chaque sommet est accessible à partir de n'importe quel autre. Autrement dit, si pour tout couple de sommets distincts
$(x_i, x_j) \in X^2$, il existe une chaîne entre x_i et x_j.

Par exemple, le graphe ci-dessous défini par les sommets {a, b, c, d} est connexe.

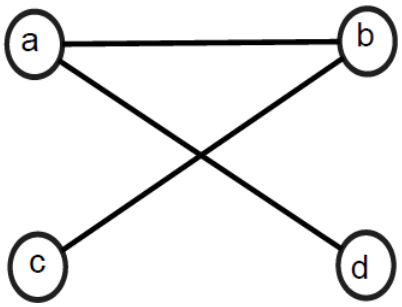


FIGURE 1.9 – Un graphe non orienté connexe

Une composante connexe d'un graphe non orienté G est un sous-graphe G_0 de G qui est connexe et maximal (c'est-à-dire qu'aucun autre sous-graphe connexe de G ne contient G_0).

Par exemple, le graphe ci-dessous est composer de deux composantes connexes :
la première est le sous-graphe défini par les sommets {a, b, c, d, e} et la seconde est le sous-graphe défini par les sommets {f, g, h}

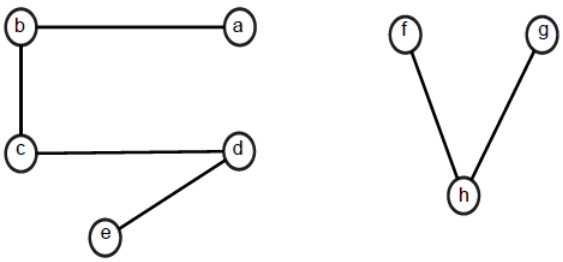


FIGURE 1.10 – Composantes connexes d'un graphe

Cas des graphes orientés :

On retrouve ces différentes notions de connexités dans les graphes orientés, en remplaçant naturellement la notion de chaîne par celle de chemin : on parle de graphe fortement connexe au lieu de connexe, de composantes fortement connexes au lieu de composantes connexes.

Plus précisément, **un graphe orienté est fortement connexe** si chaque sommet est accessible à partir de n'importe quel autre. Autrement dit, si pour tout couple de sommets distincts (x_i, x_j) $\in X^2$, il existe un chemin de x_i vers x_j, et un chemin de x_j vers x_i. Par exemple, le graphe ci-dessous est fortement connexe :

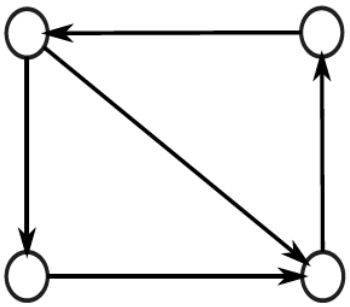


FIGURE 1.11 – Graphe fortement connexe

Une composante fortement connexe d'un graphe orienté G est un sous-graphe G_0 de G qui est fortement connexe et maximal (c'est-à-dire qu'aucun autre sous-graphe fortement connexe de G ne contient G_0).

Par exemple, le graphe ci-dessous est composer de deux composantes fortement connexes : la première est le sous-graphe défini par les sommets {a, b, c, d} et la seconde est le sous-graphe défini par les sommets {e, f, g}.

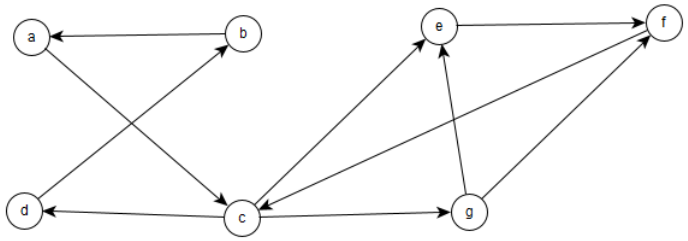


FIGURE 1.12 – Graphe fortement connexe

1.3.3 Flux et Flots

Flux

Le flux est la quantité ϕ_{ij} transportée sur chaque arc (i, j).

Flot :

Un flot φ est déterminé par la donnée du flux pour tout arc du réseau de transport. La valeur d'un flot $V(\varphi')$ est par définition, la somme des flux partant de la source $x_1(V(\varphi')$ est aussi égale à la somme des flux des arcs arrivant sur le puits $x_p)$.
La loi de Kirchoff (loi de conservation aux noeuds) est dite le flot entrant égale au flot sortant.

$$\sum_{\varphi \in w^+} v(\varphi) = \sum_{\varphi \in w^-} v(\varphi) \ \forall i \in w/\{s, t\}$$

Où le cocycle
$w^+(i)$ est l'ensemble des arcs sortant en i.
$w^-(i)$ est l'ensemble des arcs entrant en i.

1.3.4 Chaîne améliorante

μ est une chaine améliorante (ou augmentante) de s a t pour un flot admissible donne si :
-$\varphi_{ij} < c_{ij}$ pour tout arc (i, j) de μ dans le bon sens (de s vers t).
-$\varphi_{ij} > 0$ pour tout arc (i, j) de μ dans le mauvais sens (de s vers t).

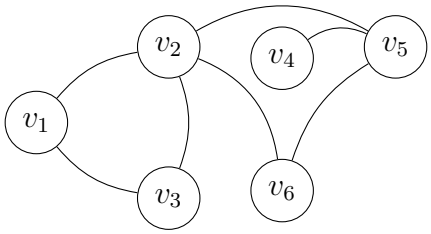
1.4 Représentations matricielle des graphes

1.4.1 Matrice d'adjacence

Soit le graphe $G = (V, E)$ d'ordre n. On suppose que les sommets de V sont numérotés de 1 à n.[11] La représentation par matrice d'adjacence de G consiste en une matrice booléenne M de taille $n * n$ telle que :

$$M_{ij} = \begin{cases} 1 & \text{si } (ij) \in E. \\ 0 & \text{sinon.} \end{cases}$$

La matrice d'adjacence associée au graphe de la figure ci-dessous est donnée par :



$$M = \begin{array}{c} \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{matrix} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix}$$

FIGURE 1.13 – graphe à 6 sommets et 7 arcs et la matrice adjacence associée

1.4.2 Matrice d'incidence

Une **matrice d'incidence** est une représentation matricielle d'un graphe montrant la relation d'incidence entre arcs et sommets.[11]

• Soit G un graphe orienté sans boucle G=(V, E) comportant n sommets $\{ v_1, v_2, ...v_n\}$,et m arcs $\{ e_1, e_2, ...e_m\}$.On appelle matrice d'incidence M=(m_{ij}) de dimension $n * m$ telle que :

$$M_{ij} = \begin{cases} 1 & \text{si } x_i \text{ est l'extrémité initiale de } e_j. \\ -1 & \text{si } x_i \text{ est l'extrémité términale de } e_j. \\ 0 & \text{si } x_i \text{ n'est pas une extrémité de } e_j. \end{cases}$$

• La matrice d'incidence d'un graphe G non orienté sans boucle définie par :

$$M_{ij} = \begin{cases} 1 & \text{si } x_i \text{ est une extrémité de } e_j, \\ 0 & \text{sinon} \end{cases}$$

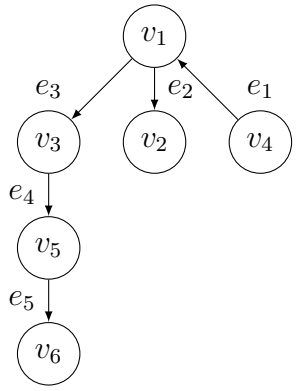
• La matrice d'incidence d'un graphe G non orienté avec boucle définie par :

$$M_{ij} = \begin{cases} 1 & \text{si } x_i \text{ est l'extrémité de } e_j. \\ 0 & \text{si } x_i \text{ n'est pas une extrémité de } e_j. \end{cases}$$

La matrice d'incidence associée au graphe de la figure ci-dessous est donnée par :

Exemple :

soit le graphe suivant :



$$M = \begin{array}{c} \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 \end{matrix} \\ \begin{pmatrix} -1 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \end{array} \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix}$$

FIGURE 1.14 – graphe à 6 sommets et 5 arêtes sans boucle et la matrice d'incidence associée

1.5 Quelques graphes particuliers

1.5.1 Graphe complet :

Un graphe $G = (V, E)$ est dit complet si tous les sommets sont deux à deux adjacents, le graphe complet à n sommet est noté K_n [7].

1.5.2 Graphe d'écart :

Lorsque l'on travaille sur les flots, il est souvent intéressant d'utiliser un graphe spécial. Soit (i, j) un arc de G de capacité minimale et maximale respective l_{ij} et u_{ij} et portant le flot x_{ij}. Il est alors possible soit :

- d'ajouter encore jusqu'à $u_{ij} - x_{ij}$ unités de flot depuis i vers j .
- de retirer jusqu'à $x_{ij} - l_{ij}$ unités de flot depuis i vers j ,ce qui peut être vu comme l'ajout d'autant d'unités de flot depuis j vers i .

Le graphe d'écart $G(x)$ va donc proposer deux arcs mettant en avant ces deux possibilités :

$$(i, j) \textit{ de capacité résiduelle } r_{ij} = u_{ij} - x_{ij} \textit{ arc direct de } (i, j)$$

$$(j, i) \textit{ de capacité résiduelle } r_{ji} = x_{ij} \textit{ arc opposé de } (i, j)$$

En outre, seuls les arcs de capacité résiduelle non nulle sont présents dans le graphe d'écart. Notez le cas intéressant où $l_{ij} = u_{ij}$. Nécessairement, tout flot compatible est tel que $x_{ij} = l_{ij} = u_{ij}$. Alors, le graphe d'écart ne contient ni l'arc (i, j) , ni l'arc (j, i) car ils ont tous deux une capacité résiduelle nulle [4].

1.5.3 Graphe biparti :

Un graphe G est dit biparti si l'ensemble de ses sommets peut-être partitionnée en deux sous-ensembles V_1 et V_2 telle que deux sommets de même sous-ensemble ne soient jamais adjacents, et on note parfois $G = (V_1, V_2, E)$ [1].
La figure 1.2 présente un graphe biparti

$V_1 = \{1, 2, 3\}$

$V_2 = \{4, 5\}$

$E = \{\{1, 4\}; \{1, 5\}\{2, 4\}; \{2, 5\}; \{3, 4\}; \{3, 5\}\}$

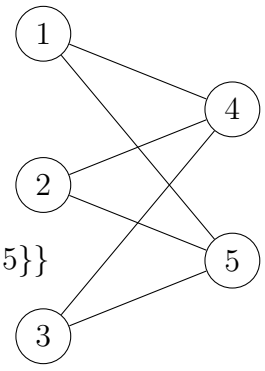


FIGURE 1.15 – Graphe biparti

Couplage

Étant donné un graphe non orienté, un couplage, est un sous-ensemble d'arêtes deux à deux disjoint . Le couplage maximum est le couplage couvrant le plus grand nombre de sommets possibles, en laissant donc le moins de sommets non saturés [19].

1.5.4 Graphe planaire :

Un graphe planaire est un graphe qu'on peut dessiner sur un plan de telle sorte que deux arêtes ne se coupent pas en dehors de leurs extrémités. Une face d'un graphe planaire est une région de plan limitée par arête (au minimum 3 arêtes). Deux face sont adjacentes si elles ont au moins une arête en commune[11].

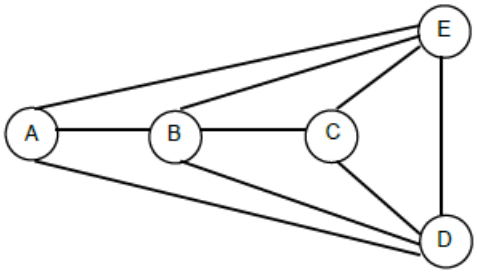


FIGURE 1.16 – Graphe planaire

1.5.5 Arbre :

Un arbre est un graphe connexe sans cycle ayant les propriétés suivantes :

- **G** est connexe et sans cycle.
- **G** est sans cycle et possède n-1 arêtes.
- **G** est connexe et admet n-1 arêtes.
- **G** est sans cycle, et en ajoutant une arête, on crée un et un seul cycle élémentaire.
- **G** est connexe, et en supprimant une arête quelconque, il n'est plus connexe.
- il existe une chaîne et une suele entre 2 sommets quelconque de **G**[11].

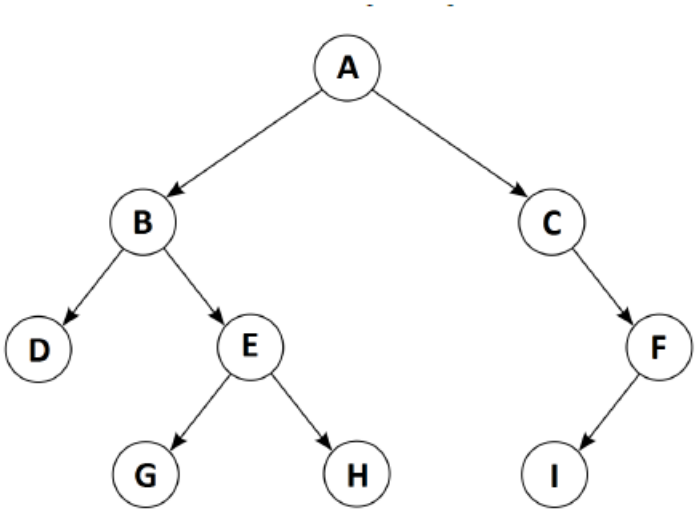


FIGURE 1.17 – Graphe planaire

Dans ce chapitre nous avons présenté les notions de base de la théorie des graphes qui seront utilisées pour la modélisation de notre problème dans les chapitres suivant

CHAPITRE 2

PROBLÈME DE TRANSPORT ET MÉTHODES DE RÉSOLUTION

Le problème du transport est un problème linéaire que peut être représenté sous forme d'un graphe et qu'on peut le résoudre en utilisant les différentes méthodes de résolution des problèmes linéaires qu'on va présenter par la suite. Dans ce chapitre nous allons présenter deux méthodes graphique (Stepping- Stone, distribution modifiée) pour la recherche de la solution optimale et les méthodes :Coin Nord-Ouest, coût minimum, approximation de vogel pour la recherche de la solution de base d'un problème de transport.

2.1 Programmation linéaire

La programmation linéaire est un outil très puissant de la recherche opérationnelle. C'est un outil générique qui peut résoudre un grand nombre de problèmes. En effet, une fois un problème modélise sous la forme d'équations linéaires, des méthodes assurent la résolution du problème de manière exacte. On distingue dans la programmation linéaire, la programmation linéaire en nombres réels, pour laquelle les variables des équations sont dans $\mathbb{R}^+$ et la programmation en nombres entiers, pour laquelle les variables sont dans $\mathbb{N}$. La résolution d'un problème avec des variables entières est nettement plus compliquée qu'un problème en nombres réels.

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

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

• La transposée A^T (aussi notée A') d'une matrice A s'obtient par symétrie axiale par rapport à la diagonale principale de la matrice. La transposée de la transposée $(A^T)^T$ est la matrice A d'origine. La transposée d'un vecteur colonne est un vecteur ligne.

2.1.1 Forme générale d'un programme linéaire

$$\begin{cases} \min o\grave{u} \max z = \sum_{j=1}^{n} c_j x_j & (1)\ . \\ \forall i = 1,..,m : \sum_{j=1}^{n} a_{ij} x_j \leq, = o\grave{u} \geq b_i & (2)\ . \\ \forall j = 1,..,n : x_{ij} \geq 0 & (3)\ . \end{cases}$$

(1) : fonction objective. .
(2) : m contraintes linéaires.
(3) : contraintes de positivité.

2.1.2 Formes matricielles classiques et conventions

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

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

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

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

2.2 La complexité algorithmique

La complexité (temporelle) d'un algorithme est le nombre d'opérations élémentaires (affectations, comparaisons, opérations arithmétiques) effectuées par un algorithme. Ce nombre s'exprime en fonction de la taille n des données.

On s'intéresse au coût exact quand c'est possible, mais également au coût moyen (que se passe-t-il si on moyenne sur toutes les exécutions du programme sur des données de taille n), au cas le plus favorable, ou bien au cas le pire. On dit que la complexité de l'algorithme est $O(f(n))$ où f est d'habitude une combinaison de polynômes, logarithmes ou exponentielles.

Ceci reprend la notation mathématique classique, et signifie que le nombre d'opérations effectuées est borné par $cf(n)$ où c est une constante, lorsque n tend vers l'infini.

Considérer le comportement à l'infini de la complexité est justifié par le fait que les données des algorithmes sont de grande taille et qu'on se préoccupe surtout de la croissance de cette complexité en fonction de la taille des données. Une question systématique à se poser est : que devient le temps de calcul si on multiplie la taille des données par 2 ? De cette façon, on peut également comparer des algorithmes entre eux.

2.2.1 La notation O(.)

La théorie de la complexité (algorithmique) vise à répondre à ces besoins. Elle permet :

1. de classer les problèmes selon leur difficulté ;
2. de classer les algorithmes selon leur efficacité ;
3. de comparer les algorithmes sans devoir les implémenter.

Comme on l'a vu dans la section précédente, les calculs à effectuer pour évaluer le temps d'exécution d'un algorithme peuvent parfois être longs et pénibles. De plus, le degré de précision qu'ils requièrent est souvent inutile. On aura donc recours à une approximation de ce temps de calcul, représentée par la notation O(.) .

Définition :

Une fonction $f(n)$ est en $O(g(n))$ si :

$$\exists n_0 \in \mathbb{N}, \exists c \in \mathbb{R}, \forall n \geq n_0 : |f(n)| \leq c|g(n)|$$

.

Autrement dit : $f(n)$ est en $O(g(n))$ s'il existe un seuil à partir duquel la fonction $f(.)$ est toujours dominée par la fonction $g(.)$ à une constante multiplicative fixée près. Les valeurs absolues importent dans la définition, mais on verra qu'en pratique, on ne devra jamais s'en soucier ; les fonctions qui nous intéresseront mesureront toujours le temps d'exécution d'un algorithme, qui sera forcément positif.

2.2.2 Calcule de la complixité d'un algorithme

la notation O(.) va nous permettre de quantifier l'efficacité d'un algorithme sous certaines hypothèses :

Définition **la complexité** d'un algorithme est la mesure **asymptotique** de son temps d'exécution **dans le pire des cas** .Il s'exprime a l'aide de la notation O(.) en fonction de la taille des données reçues en entrée .

Les deux précisions sur le caractère de cette mesure important :

1. Asymptotique signifie que l'on s'intéresse à des données très grandes ; en effet ,les petites valeurs ne sont pas assez informative .
2. "dans le pire des cas " signifie que l'on s'intéresse à la performance de l'algorithme dans les situations où le problème prend le plus de temps à résoudre ; on veut être sur que l'algorithme ne prendra jamais plus de temps que ce qu'on a estimé .

Les bases

Le calcul de la complexité d'un algorithme s'effectue de manière similaire à celui du temps d'exécution, si ce n'est qu'on remplace maintenant les unités de temps par les approximations fournies par la notation O(.) . Plus précisément :

- chaque instruction basique (affectation d'une variable, comparaison, +,/ ,*,.....) consomme un temps constant représenté par la notation O(1) ;
- chaque itération d'une boucle rajoute la complexité de ce qui est effectué dans le corps de cette boucle ;
- chaque appel de fonction rajoute la complexité de cette fonction ;
- pour obtenir la complexité de l'algorithme, on additionne le tout.

On aura aussi recours aux simplifications suivantes :

1. on oublie les constantes multiplicatives (elles valent 1) ;
2. on annule les constantes additives ;
3. on ne retient que les termes dominants.

2.3 Positionnement du problème

Dans le problème des transports, nous avons certaines origines, Une usine représentant nos articles de production et nos quantités d'approvisionnement Le produit souhaité atteint plusieurs destinations. Cela doit commencer à partir de Afin de maximiser les profits ou de minimiser les coûts. Nous avons donc Pris la production comme origine et l'approvisionnement comme destination.

2.4 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 :

- 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 $i \in \{1, 2, 3..., m\}$ et $j \in \{1, 2, 3..., n\}$.

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

2.4.1 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 $i \in \{1, 2, 3...m\}$ et $j \in \{1, 2, 3, ...n\}$.

2.4.2 Fonction objective

Le problème consiste à déterminer les quantités x_{ij} à transporter de façon que le coût total de transport $\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$ 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[12].

2.4.3 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 : $\sum_{j}^{n} x_{ij} = a_i, i \in \{1, 2, 3...m\}$.
- La demande à chaque destination doit être satisfaite : $\sum_{i}^{m} x_{ij} = b_j, j \in \{1, 2, 3...n\}$.
- La non négativité des quantités :$x_{ij} \geq 0, \forall i \in \{1, ..., m\}; \forall j \in \{1, ..., n\}$

2.4.4 Formulation mathématique

$$\begin{cases} \min z = \sum_{i}^{m}\sum_{j}^{n} c_{ij}x_{ij} \quad . \\ \quad \sum_{j}^{n} x_{ij} = a_i \qquad \forall i \in \{1, ..., m\} \ . \\ \quad \sum_{i}^{m} x_{ij} = b_j \qquad \forall j \in \{1, ..., n\}. \\ \quad a_i \in \mathbb{R}^+ \qquad \forall i \in \{1, ..., m\} \ . \\ \quad b_i \in \mathbb{R}^+ \qquad \forall j \in \{1, ..., n\}. \\ \quad x_{ij} \in \mathbb{N}^+ \qquad \forall (i, j) \in \{1, ..., m\} \times \{1, ..., n\}. \end{cases}$$

Il s'agit d'un programme linéaire avec $m * n$ variables de décision, $m + n$ contraintes fonctionnelles et $m \times n$ contraintes non négatives.
m : Nombre de sources.
n : Nombre de destinations.
a_i : Disponibilité de la $i^{ème}$ source.
b_j : Demande de la $j^{ème}$ destination .

c_{ij} : Coût unitaire de transport de la $i^{ème}$ source à la $j^{ème}$ destination .
x_{ij} : Quantité transportée de la $i^{ème}$ source à la $j^{ème}$ destination .[15]

Une condition nécessaire et suffisante pour l'existence d'une solution réalisable au problème transport est que : $\sum_i^m a_i = \sum_j^n b_j, \forall (i,j) \in \{1,...m\} * \{1,...n\}$.

2.5 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.[14]

la figure 3.1 présente un tableau de problème de transport :

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

FIGURE 2.1 – Tableau de transport

2.6 Réseau de transport

Graphiquement, le problème du transport est souvent visualisé comme un réseau avec m noeuds sources, n noeuds destinations et un ensemble de $m \times n$ "arcs " Ceci est représenté dans la figure 3.2

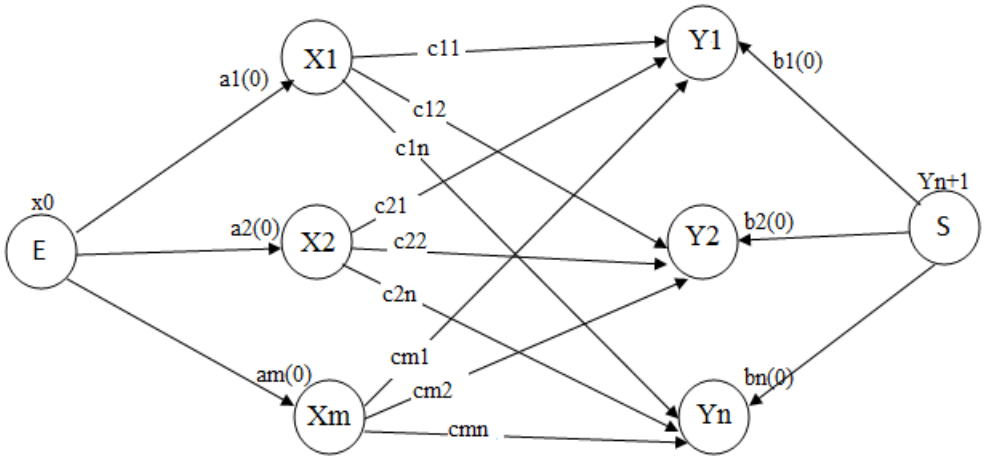


FIGURE 2.2 – Réseau de transport

2.7 Dégénérescence en problème de transport :

La dégénérescence existe dans un problème de transport lorsque le nombre de cellules remplies est inférieur $(m + n - 1)$. 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[14] .

2.8 Structure de la résolution de problème de transport :

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

De telle sorte que les exigences des destinations soient satisfaites. Cela peut être réalisé soit par une inspection, soit par des règles simples. Nous commençons par imaginer que la table de transport est vide, c'est-à-dire initialement tout $x_{ij} = 0$. Les procédures les plus simples pour l'allocation initiale seront discutées dans la section suivante.[2]

2.8.1 Solution de base réalisable

Définition

On appelle solution de base réalisable 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.

Si le nombre d'allocations dans les solutions de bases réalisables est inférieur à $(m+n-1)$, 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 $\varphi_{ij} = \min\{a', b'\}$ $(a'_i, b'_j \neq 0, \varphi_{ij} \in \mathbb{N})$ 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 $(m = n - 1)$ flux φ_{ij} non nuls et nous obtenons une solution de base :
$(m \times n - (m + n - 1) = (m - 1)(n - 1))$ flux nuls.

2.8.2 Solution optimale

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

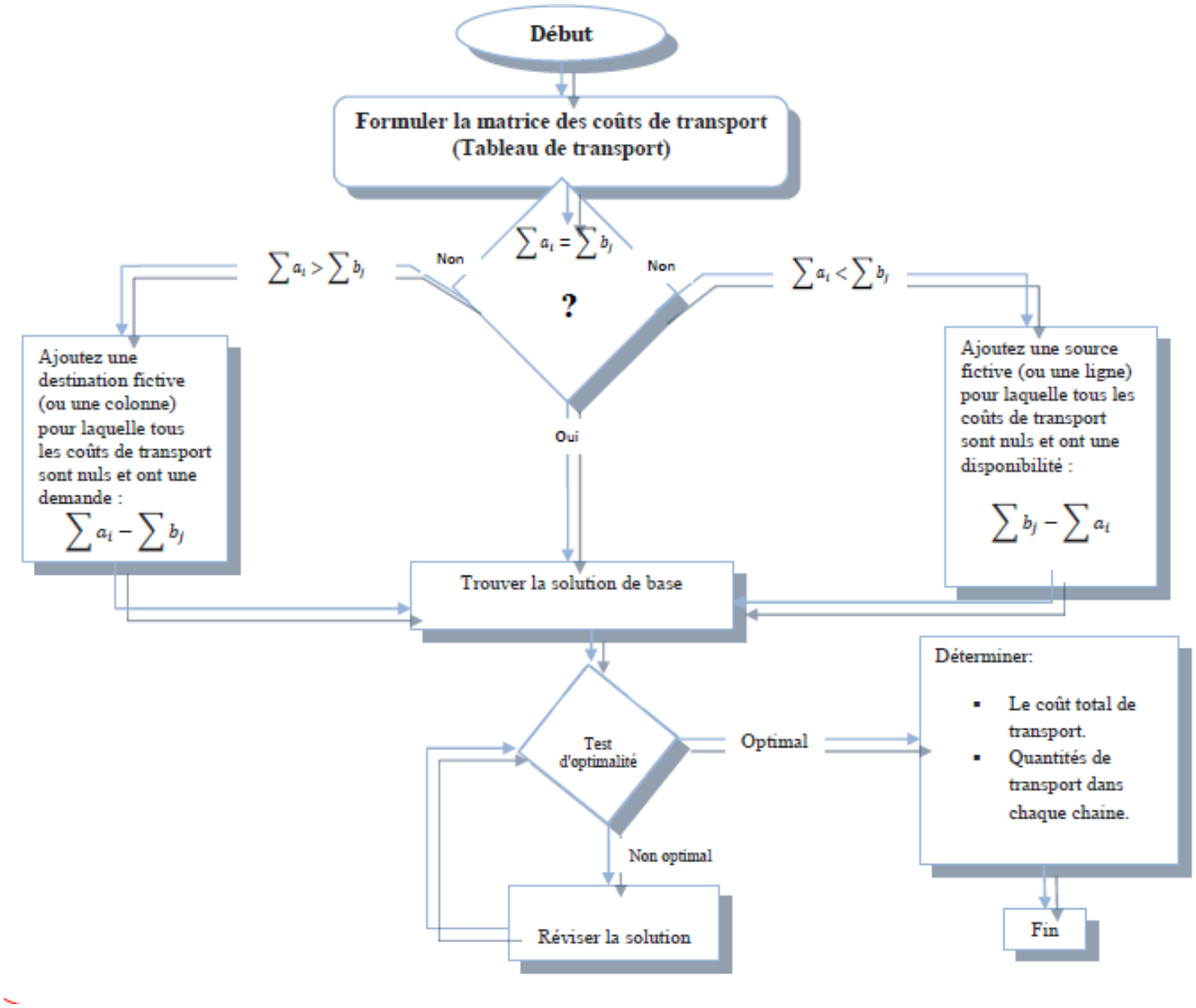


FIGURE 2.3 – Organigramme de la résolution de problème de transport.

2.8.3 Organigramme de résolution pour le problème de transport

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

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 une source fictive ou une destination fictive 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, revenir 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.

2.8.4 Algorithme général de résolution de problème de transport

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

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

Étape 1 : Formuler et configurer le problème sous la forme matricielle : La formulation du problème de transport est similaire à la formulation du problème PL. Ici, la fonction objective est le coût total du transport et les contraintes sont l'offre et la demande disponibles à chaque source et destination, respectivement.

Étape 2 : Obtenir une première solution de base réalisable : Cette solution de base initiale peut être obtenue en utilisant l'une des méthodes suivantes :

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

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

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

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

Étape 3 : Tester la solution de base initiale pour l'optimalité : L'utilisation de l'une des méthodes suivantes pour tester l'optimalité de la solution de base initiale obtenue :

- Méthode Stepping Stone.
- Méthode de distribution modifiée.
- 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.

2.9 Méthode de détermination de la solution de base

2.9.1 Méthode du COIN NORD-OUEST

Présentation

La méthode du coin nord-ouest est une méthode facile mais elle n'a pas de sens économique. Puisqu'elle consiste à affecter au coin nord-ouest de chaque grille la quantité maximum possible sans se préoccuper de l'importance du coût.[6]

Principe

Étape 1 : Localiser la cellule $(p;q)$ qui se trouve dans le coin nord-ouest, c'est à-dire en haut à gauche, de la partie non-éliminée du tableau de transport.

Étape 2 : Envoyer le maximum d'unités pour la cellule $(p;q)$.Ainsi x_{pq} est initialisé comme étant le $\min\{a_p, b_q\}$. Ajuster ensuite a_p et b_q , en tenant compte du montant x_{pq} à expédier. Exprimons cette phrase à l'aide d'inégalités : $x_{pq} = \min\{a_p, b_q\}$

$$a'_p = a_p - x_{pq}$$

$$b'_p = b_p - x_{pq}$$

Entourer (ou mettre en évidence d'une autre manière) le coût c_{pq} .À la fin de cette étape, soit a'_p, soit b'_q est nul, soit les deux.

Étape 3 :

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

Étape 4 :

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

• Appliquons la méthode de Coin nord-ouest au problème de transport donné par la table 3.1.

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

TABLE 2.1 – Problème de transport initial

Itération 1 :

Étape 1 Le tableau restant se compose de deux lignes et de trois colonnes ; son coin nord-ouest est la cellule $(1;1)$, x_{11} entre donc dans la base.

Étape 2

$$x_{11} = min\{a_1; b_1\} = min\{400; 300\} = 300$$
$$a'_1 = a_1 - x_{11} = 400 - 300 = 100$$
$$b'_1 = b_1 - x_{11} = 300 - 300 = 0$$

–

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.

Origines \ Destinations	~~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

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

$$a'_1 = a_1 - x_{12} = 100 - 100 = 0$$
$$b'_2 = b_2 - x_{12} = 200 - 100 = 100$$

Étape 3 $a'_1 = 0$ et $b'_2 = 100$; 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	17 100	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.

2

Origines \ Destinations	1	2	3	Offre
1	26 300	17 100	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

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

2.9.2 Méthode de coût minimum

Définition

La méthode du Coût Minimum est une méthode pour calculer une solution de base réalisable d'un problème de transport où les variables de base sont choisies en fonction du coût unitaire du transport. La méthode du coût minimum trouve une meilleure solution de départ en se concentrant sur les coûts de transport les moins chers.[6]

Principe

cette méthode ne diffère de la précédente que par le critère appliqué à l'étape (1), exposée ici , les étapes (2), (3) et (4) restant les mêmes.

Étape 1 Trouver la cellule $(p; q)$, telle que c_{pq} est le plus petit coût de tout le tableau.

– 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.

Appliquons la méthode le cout minimum au problème de la table 3.1
Itération 1 :

Étape 1 La cellule $(p; q)$ choisie est la cellule (2 ; 3) dont le coût (10) est le plus petit de l'ensemble du tableau.

Étape 2 Calculons x_{23}, a'_2 et b'_3.

$$x_{23} = min\{a_2; b_3\} = min\{500; 400\} = 400$$
$$a'_2 = a_2 - x_{23} = 500 - 400 = 100$$
$$b'_3 = b_3 - x_{23} = 400 - 400 = 0$$

Étape 3 Comme $a'_2 = 100$ et $b'_3 = 0$, il faut éliminer la colonne 3.

Étape 4 Il reste deux lignes (1 et 2)et deux colonnes (1 et 2),il faut donc choisir un nouvel x_{pq} qui entrera à son tour dans la solution réalisable de base

le tableau après cette première itération.

Origines \ Destinations	1	2	~~3~~	Offre
1	26	15	~~14~~	400
2	24	18	~~10~~ 400	500 100
Demande	300	200	~~400~~ 0	900

Reprenons nos calculs à l'étape (1) avec un tableau réduit aux lignes 1 et 2 et aux colonnes 1 et 2 .

Itération 2 :

Étape 1 Le coût minimum sur ce tableau est 15, soit celui de la cellule (1 ; 2) ,on va donc initialiser x_{12}.

Étape 2

$$x_{12} = min\{a_1; b_2\} = min\{400; 200\} = 200$$
$$a'_1 = a_1 - x_{12} = 400 - 200 = 200$$
$$b'_2 = b_2 - x_{12} = 200 - 200 = 0$$

Étape 3 $a'_1 = 200$ et $b'_2 = 0$, il faut donc éliminer la colonne 2.

Étape 4 Comme il reste deux lignes et une colonne, nous n'avons pas terminé,nous cherchons alors le troisième x_{pq} à entrer dans la base.
le tableau après cette deuxième itération.

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

Itération 3 :

Étape 1 Le coût minimum sur la ligne 1, la ligne 2 et la colonne 1 est 25, soit le coût de la cellule (2 ; 1).

Étape 2

$$x_{21} = min\{a_2; b_1\} = min\{200; 100\} = 200$$
$$a'_2 = a_2 - x_{21} = 200 - 200 = 0$$
$$b'_1 = b_1 - x_{21} = 300 - 200 = 100$$

Étape 3 $a'_2 = 0$ et $b'_1 = 100$, on élimine donc la ligne 2.

Étape 4 Il reste une ligne (1) et une colonne (1), il faut donc procéder à une quatrième itération.

le tableau après cette troisième itération.

Origines \ Destinations	1	2	3	Offre
1	26	15 200	14	400 200
2	24 100	18	14 400	500 0
Demande	300 200	200 0	400 0	900

Itération 4 :

Étape 1 Le seul x_{pq} qui peut encore entrer dans la base est x_{11} .

Étape 2

$$x_{11} = min\{a_1; b_1\} = min\{200; 200\} = 200$$
$$a'_1 = a_1 - x_{11} = 200 - 200 = 0$$
$$b'_1 = b_1 - x_{11} = 200 - 200 = 0$$

Étape 3 $a'_1 = 0$ et $b'_1 = 0$, la ligne 1 est la dernière des lignes, on élimine donc la colonne 1.

Étape 4 Il ne reste que la ligne 1, nous avons donc terminé.

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

À l'aide de cet exemple, nous pouvons vérifier les affirmations formulées précédemment. Tout d'abord, en raison de la redondance, on trouve une solution réalisable de base initiale avec $(m + n - 1)$ variables, ici $2 + 3 - 1 = 4$. Ensuite, à chaque itération, une seule contrainte est satisfaite, ce qui se voit facilement sur le tableau.
La dernière contrainte est automatiquement satisfaite à la dernière itération ; donc, si après la quatrième itération l'une des origines ou des destinations avait eu une valeur non-nulle, cela aurait signifié qu'il n'existe pas de solution réalisable.
Nous pouvons maintenant calculer le coût de transport total :

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

Il ne s'agit pas encore du coût minimum ; il sera déterminé lors de la recherche de la solution réalisable optimale.

Méthode d'Approximation de Vogel

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.[12]

La méthode de Vogel 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 maximum 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.

Remarques :

- La Méthode d'Approximation de Vogel (VAM) est connue aussi par :l'heuristique de Balas-Hammer /la Méthode de la différence maximum / la méthode de pénalité unitaire /la méthode des regrets maximaux successifs.
- La méthode VAM est basée sur la notion de coût de pénalité où de regret.
- Un coût de pénalité est la différence entre le coût de case le plus grand et le plus important d'une rangée (ligne où colonne).
- Dans VAM, la première étape consiste à développer un coût de pénalité pour chaque source et destination.[16]
- Le coût de pénalité est calculé en soustrayant le coût de case minimum du coût de case supérieur suivant dans chaque rangée.

L'algorithme d'Approximation de Vogel

Δ_l représente la différence entre le coût minimum et celui immédiatement supérieur sur une ligne. Δ_c représente la différence entre le coût minimum et celui immédiatement supérieur sur une colonne.

1. Calculer les différences Δ_l et Δ_c pour chaque ligne et colonne.
2. Sélectionner la ligne ou la colonne ayant le Δ_l ou Δ_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 demande soit en disponibilité.
6. Éliminer la ligne ou la colonne ayant sa disponibilité ou demande satisfaite.
7. SI nombre de lignes ou colonnes > 2 retour en 2. SINON affecter les quantités restantes aux liaisons.

• Appliquons la méthode de d'approximation de Vogel au problème de la table 3.1

Itération 1 :

Étape 1

- La différence entre le coût de la ligne 1 $\triangle_l = 15 - 14 = 1$
- La différence entre le coût de la ligne 2 $\triangle_l = 18 - 10 = 8$
- La différence entre le coût de la colonne 1 $\triangle_c = 26 - 25 = 1$
- La différence entre le coût de la colonne 2 $\triangle_c = 18 - 15 = 3$
- La différence entre le coût de la colonne 3 $\triangle_c = 14 - 10 = 4$

Étape 2

$$max\{\triangle_{c1}, \triangle_{c2}, \triangle_{c3}\} = max\{1, 3, 4\} = 4$$

on sélection la colonne 3.

Étape 3

Le coût le plus faible da la colonne 3

$min\{14, 10\} = 10$ donc le coût le plus faible 10 de la ligne 2.

Étape 4

$$x_{23} = min\{a_2; b3\} = min\{500; 400\} = 400$$
$$a'_2 = a_2 - x_{23} = 500 - 400 = 100$$
$$b'_3 = b_3 - x_{23} = 400 - 400 = 0$$

Étape 5 $b'_3 = 0$, on élimine donc la colonne 3

Étape 6 Il reste une ligne et deux colonne, nous n'avons pas terminé, on revient à l'étape 1.

Destinations / Origines	1	2	~~3~~	Offre	Δ_l
1	26	15	~~14~~	400	3
2	25	18	~~10~~ 400	500 100	4
Demande	300	200	~~400~~ 0	900	
Δ_c	1	3	4		

Itération 2 :

Étape 1

-La différence entre le coût de la ligne 1 $\triangle_l = 26 - 15 = 11$

- La différence entre le coût de la ligne 2 $\triangle_l = 25 - 18 = 7$

- La différence entre le coût de la colonne 1 et 2 il reste le même.

Étape 2

$$max\{\triangle_{c1}, \triangle_{c2}\} = max\{1, 3\} = 3$$

on sélection la colonne 2.

Étape 3

Le coût le plus faible da la colonne 2

$min\{15, 18\} = 15$ donc le coût le plus faible 15 de la ligne 1.

Étape 4

$$x_{21} = min\{a_2; b_1\} = min\{400; 200\} = 200$$
$$a'_2 = a_2 - x_{21} = 400 - 200 = 200$$
$$b'_1 = b_1 - x_{21} = 200 - 200 = 0$$

Étape 5 $b'_2 = 0$, on élimine donc la colonne 2

Origines \ Destinations	1	2	3	Offre	Δ_l
1	26	~~15~~ 200	~~14~~	400	11
2	25	~~18~~	~~14~~ 400	~~500~~ 0	7
Demande	300	~~200~~ 0	~~400~~ 0	900	
Δ_c	1	3			

Étape 6 Le nombre de colonne inférieure à 2 donc affecter les quantités restantes aux liaison $x_{11} = 200$ et $x_{21} = 100$,nous avons donc terminé.

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

Nous pouvons maintenant calculer le coût de transport total :

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

On obtient par hasard la même solution réalisable de base initiale qu'avec la deuxième méthode proposée. Ceci est à considérer comme une exception due aux dimensions réduites du problème choisi comme exemple.

2.9.3 Méthode des pénalité (Balas-hammer)

Cette méthode est basée sur le calcul de la différence entre deux valeurs (le regret). le regret associé a une ligne ou une colonne est la différence entre le cout minimum et le cout 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 cout minimum [12].

Algorithme d'approximation de Balas-hammer

Δ_l représente la différence ente le cout minimum et celui immédiatement supérieur sur une ligne.
Δ_c représente la différence ente le cout minimum et celui immédiatement supérieur sur une colonne.
Étape 1 : Calculer les différences Δ_l et Δ_c pour chaque ligne et colonne.
Étape 2 : Sélectionner la ligne ou la colonne ayant le Δ_l ou Δ_c maximum.
Étape 3 : Choisir dans cette ou colonne le cout le plus faible.
Étape 4 : Attribuer à la résolution (i, j) correspondante le maximum possible de matière transportable de façon à saturer soit la destination soit la disponibilité.
Étape 5 : Calcul la quantité résiduelle soit en demande soit en disponibilité.
Étape 6 : Éliminer la ligne ou la colonne ayant sa disponibilité ou demande satisfaite.
Étape 7 : SI nombre de ligne ou colonnes > 2 retour en 2,**SI NON** affecter les quantités restantes aux liaisons.

application de la méthode de Balas-hammer

Nous considérons le tableau de problème de transport initial 3.1

Itération 1 :
Étape 1
- La différence entre le coût de la ligne 1 $\triangle_l = 15 - 14 = 1$
- La différence entre le coût de la ligne 2 $\triangle_l = 18 - 10 = 8$
- La différence entre le coût de la colonne 1 $\triangle_c = 26 - 25 = 1$
- La différence entre le coût de la colonne 2 $\triangle_c = 18 - 15 = 3$
- La différence entre le coût de la colonne 3 $\triangle_c = 14 - 10 = 4$

Étape 2

$$max\{\triangle_{c1}, \triangle_{c2}, \triangle_{c3}, \triangle_{l1}, \triangle_{l2}\} = max\{1, 3, 4, 1, 8\} = 8$$

on sélection la ligne 2.

Étape 3
Le coût le plus faible de la ligne 2

donc le coût le plus faible est 10 .
Étape 4
– Attribuer a x_{23} correspondante le maximum possible de façon a saturer soit la destination $[x_{23} = min\{a_2; b3\} = min\{500; 400\} = 400]$

$$a'_2 = a_2 - x_{23} = 500 - 400 = 100$$
$$b'_3 = b_3 - x_{23} = 400 - 400 = 0$$

Étape 5 calcule la quantité résiduelle soit en demande soit en disponibilité on vas donc élimine donc la colonne 3.

Étape 6 Il reste deux ligne et deux colonne, nous n'avons pas terminé, on revient à l'étape 1.

Origines \ Destinations	1	2	3	Offre	Δ_l
1	26	15	~~14~~	400	3
2	25	18	~~10~~ 400	500 100	4
Demande	300	200	~~400~~ 0	900	
Δ_c	1	3	4		

Itération 2 :

Étape 1

-La différence entre le coût de la ligne 1 $\triangle_l = 26 - 15 = 11$

- La e différence entre le coût de la ligne 2 $\triangle_l = 25 - 18 = 7$

- La différence entre le coût de la colonne 1 $\triangle_{c1} = 26 - 25 = 1$

- La différence entre le coût de la colonne 2 $\triangle_{c2} = 18 - 15 = 3$

Étape 2

$$max\{\triangle_{l1}, \triangle_{l2}, \triangle_{c1}, , \triangle_{c2}\} = max\{11, 7, 1, 3\} = 11$$

on sélection la ligne 1.

Étape 3

Le coût le plus faible da la ligne 1

donc le coût le plus faible 15 de la colonne 2.

Étape 4

Attribuer a x_{12} correspondante le maximum possible de façon a saturer soit la destination

$$x_{21} = min\{a_2; b_1\} = min\{400; 200\} = 200$$

$$a'_2 = a_2 - x_{21} = 400 - 200 = 200$$
$$b'_1 = b_1 - x_{21} = 200 - 200 = 0$$

Étape 5 calcule la quantité résiduelle soit en demande soit en disponibilité et on élimine donc la colonne 2

Origines \ Destinations	1	2	~~3~~	Offre	Δ_l
1	26	~~15~~ 200	~~14~~	400	11
2	25	~~18~~	~~14~~ 400	~~500~~ 0	7
Demande	300	~~200~~ 0	~~400~~ 0	900	
Δ_c	1	3			

Étape 6 Le nombre de colonne inférieure à 2 donc affecter les quantités restantes aux liaison $x_{11} = 200$ et $x_{21} = 100$,nous avons donc terminé.

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

Nous pouvons maintenant calculer le coût de transport total :

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

On obtient par hasard la même solution réalisable de base initiale qu'avec la deuxième méthode proposée. Ceci est à considérer comme une exception due aux dimensions réduites du problème choisi comme exemple.

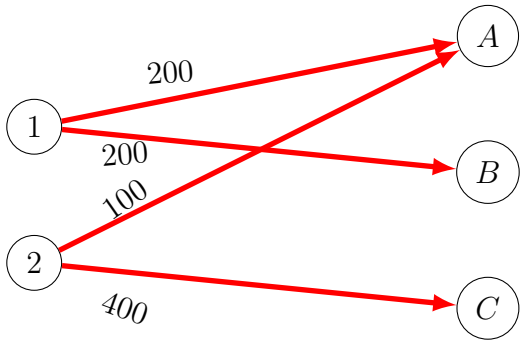


FIGURE 2.4 – Graphe d'affectation de la méthode du Vogel et Balas-Hammer

2.10 Méthode d'optimisation de la solution de base

2.10.1 Méthode de Stepping-Stone

L'algorithme du Stepping-Stone est un algorithme itératif (par étapes successives) visant à améliorer (faire baisser le coût global) une solution de base.[6]

Il nous faut donc une solution de départ pour démarrer l'algorithme.

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

On appelle :
$T_i \ll u_i \gg$: potentiel origine.
$T_j \ll v_j \gg$: potentiel destination.
∂_{ij} : coût marginal de la liaison (x_i, x_j).

Algorithme de stepping stone

1. Déterminer une solution de base initiale.
2. Calculer les coûts marginaux $\delta_{ij} = c_{ij} - (t_j t_i)$ avec $t_j - t_i = \theta$, I_j la tension de l'arc (i, j), ti s'appelle le potentiel du sommet i de l'arc (i, j). Si tous les coûts marginaux sont positifs où nuls alors FIN. La solution est optimal, Sinon passer à 3.
3. Pour tous les coûts marginaux négatifs, chercher la chaine de substitution et déterminer la quantité maximum qui peut être déplacé et passer à (4). Alors le gain correspondant est égale au produit de cette quantité par le coût marginale.
4. Retenir la chaine de substitution qui réalise la plus grande diminution du coût de trans-

port, l'effectuer et revenir à (2).

Remarque : Le coût marginale : La quantité (positive ou négative) qui s'ajoute au coût globale lorsqu'on veut transporter une unité sur un arc de flux nul.

Comment déterminer les potentiels ?

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

Comment déterminer les δ ?

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

Comment déterminer les quantités à transporter(q) ?

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

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.

• Appliquons la méthode de Stepping-Stone au problème de la table 3.1

Étape 1 Tableau de transport initial (table 3.1).

Étape 2 À l'aide de la méthode du coin nord-ouest, nous obtenons la solution réalisable de base initiale suivante :

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

$$v(1;3) = c_{13} - c_{23} + c_{22} - c_{12} = 14 - 10 + 18 - 15 = 7.$$
$$v(2;1) = c_{21} - c_{11} + c_{12} - c_{22} = 25 - 26 + 15 - 18 = -4$$

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.

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.

Revenons à notre solution ; nous remarquons qu'elle est la même que la solution réalisable de base initiale trouvée, soit par la méthode coût minimum, soit par la méthode Vogel, pour lesquelles z = 14700. Ceci montre, sur ce petit problème, que ces deux méthodes sont plus efficaces que celle du coin nord-ouest.Il est bien clair que pour des problèmes plus grands que celui envisagé ici, ni la méthode de la matrice minimale, ni la méthode Vogel ne donnent directement la solution réalisable de base optimale.

2.10.2 Méthode de Distribution Modifiée

Définition

La Méthode de distribution modifiée (où des penalties) : est une version modifiée de la méthode de stepping stone dans laquelle les équations mathématiques remplacent les chaines de substitutions. Cette méthode est plus pratique que stepping stone. En appliquant la méthode MODI, nous commençons par une solution initiale obtenue en utilisant les méthodes citées à la section précédente. Ensuite, nous devons calculer une valeur ui pour chaque ligne i et v_j pour chaque colonne j dans la table de transport.[2]

Les étapes de la méthode de distribution modifiée

1. Pour calculer les valeurs u_i et v_j pour chaque ligne et chaque colonne, réduire les équations : $u_i + v_j = c_{ij}$.
2. Une fois que toutes les équations ont été écrites, définissez l'une des deux variables u_i ou v_j à zéro, et résolvez le système d'équations pour toutes les valeursu_i et v_j.
3. Calculez l'indice d'amélioration $\triangle_{ij}$ pour chaque cellule inutilisée par l'amélioration de la formule :$\triangle_{ij} = c_{ij} - u_i - v_j$.
4. Transférer la plus grande quantité possible à la cellule qui a $\triangle_{ij}$ le plus négatif en créant un cycle qui satisfait la demande et la disponibilité de chaque rangé.
5. Répétez les étapes 2 à 4 jusqu'à ce qu'il n'y ait pas de $\triangle_{ij}$ négatif.
6. Calculez le coût total en multipliant chaque allocation (x_{ij}) par son spécifique coût (c_{ij}).

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

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

• Appliquons la méthode de distribution modifiée au problème suivant :

	v_j	v_1	v_2	v_3	
u_i	Destinations / Origines	1	2	3	Offre
u_1	1	26 300	15 100	14	400
u_2	2	25	18 100	10 400	500
	Demande	300	200	400	900

Itération 1 :

Étape 1 Calculer pour toutes les cellules allouées :$u_i + v_j = c_{ij}$: Coût de transport unitaire pour la case ij.

$x_{11} : u_1 + v_1 = 26$

$x_{12} : u_1 + v_2 = 15$

$x_{22} : u_2 + v_2 = 18$

$x_{23} : u_2 + v_3 = 10$

Étape 2 On met donc $u_1 = 0$ et on obtient :

$v_1 = 26$, $v_2 = 15$, $u_2 = 3$, $v_3 = 7$

Étape 3 Utilisez la formule de penalties pour évaluer toutes les cellules vides :

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

$x_{13} : \triangle_{13} = c_{13} - u_1 - v_3 = 14 - 0 - 7 = 7$

$x_{21} : \triangle_{21} = c_{21} - u_2 - v_1 = 25 - 3 - 26 = -4$

Étape 4 1. $x'_{21} = min\{x_{11}, x_{22}\} = min\{300, 100\} = 100$

2. $x_{12} = x_{12} - x'_{12} = 100 + 100 = 200$

   $x'_{11} = x_{11} - x'_{21} = 300 - 100 = 200$

   $x'_{22} = x_{22} - x'_{21} = 100 - 100 = 0$

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

	v_j	v_1	v_2	v_3	
u_i	Destinations / Origines	1	2	3	Offre
u_1	1	26 200	15 200	14	400
u_2	2	25 100	18	10 400	500
	Demande	300	200	400	900

Itération 2 :

Étape 1

$x_{11} : u_1 + v_1 = 26$

$x_{12} : u_1 + v_2 = 15$

$x_{21} : u_2 + v_1 = 25$

$x_{23} : u_2 + v_3 = 10$

Étape 2 On met donc $u_1 = 0$ et on obtient :

$v_1 = 26$, $v_2 = 15$, $u_2 = -1$, $v_3 = 11$.

Étape 3

$x_{13} : \triangle_{13} = c_{13} - u_1 - v_3 = 14 - 0 - 11 = 3$

$x_{22} : \triangle_{22} = c_{22} - u_2 - v_2 = 18 + 1 - 11 = 8$

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

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

2.11 Problème d'affectation

le problème d'affectation est un problème classique de recherche opérationnelle peut être résolu en temps polynomial par l'algorithme hongrois, il appartient par conséquent a la classe

de complexité P.

2.11.1 Résolution du problème d'affectation

Algorithme hongrois

Soit $V = (v_{ij})1 \leq i,j \leq n$ la matrice des coûts associée à un problème d'affectation de coût minimal .

- Phase 1 : Réduction initiale
  1. Soustraire l'élément minimum de la ligne i de chaque élément de la ligne i,pour tout $i = 1, ..., n$.
  2. Soustraire l'élément minimum de la colonne j de chaque élément de la colonne j, pour tout $j = 1, ..., n$.
- Phase 2 : Marquage des zéros Prendre la ligne contenant le moins de zéros ; encadrer le premier zéro de cette ligne et barrer les autres zéros de la ligne et de la colonne du zéro encadré. Refaire cette opération jusqu'à impossibilité d'encadrer un zéro.
- Phase 3 : Recherche d'une solution optimale
  1. procédure de marquage des lignes et des colonnes :
     - Marquer les lignes ne contenant aucun 0 encadré (s'il n'y en a pas : FIN) ;
     - Marquer toute colonne qui a un 0 barré sur une ligne marquée ;
     - Marquer toute ligne qui a un 0 encadré dans une colonne marquée ;
     - Retour à b et c jusqu'à ce qu'il n'y ait plus de ligne ou de colonne à marquer.
  2. Rayer chaque ligne non marquée et chaque colonne marquée.
  3. Réduction : choisir le plus petit élément p du tableau non rayé, l'ajouter aux colonnes non rayées et le soustraire aux lignes rayées.
  4. Retour à phase 2.

Considérons le problème d'affectation suivant reste a affecté chaque poste a un seul ingénieur et chaque ingénieur a un seul poste donné par la table suivante :

17	15	9	5	12
16	16	10	5	10
12	15	14	11	5
4	8	14	17	13
13	9	8	12	17

la première itération

Phase 1 : Réduction initiale

					min
17	15	9	5	12	5
16	16	10	5	10	5
12	15	14	11	5	5
4	8	14	17	13	4
13	9	8	12	17	8

=Réduction des lignes⇒

12	10	4	0	7
11	11	5	0	5
7	10	9	6	0
0	4	10	13	9
5	1	0	4	9

	12	10	4	0	7
	11	11	5	0	5
	7	10	9	6	0
	0	4	10	13	9
	5	1	0	4	9
min	0	1	0	0	0

=Réduction des colonnes⇒

12	9	4	0	7
11	10	5	0	5
7	9	9	6	0
0	3	10	13	9
5	0	0	4	9

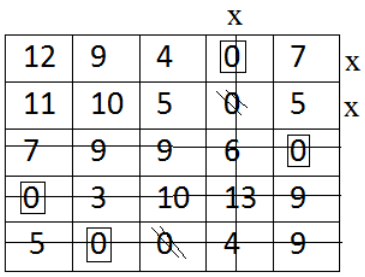
Phase 2 : marquage des zéros

12	9	4	[0]	7
11	10	5	~~0~~	5
7	9	9	6	[0]
[0]	3	10	13	9
5	[0]	~~0~~	4	9

Phase 3 : recherche d'une solution optimale

1) Marquage des lignes et colonnes

			X		
12	9	4	[0]	7	X
11	10	5	~~0~~	5	X
7	9	9	6	[0]	
[0]	3	10	13	9	
5	[0]	~~0~~	4	9	



2) Rayure des lignes et colonnes

3) Réduction

8	5	0	0	3
7	6	2	0	1
7	9	9	10	0
0	3	10	17	9
5	0	0	8	9

deuxième itération

Phase 2 : marquage des zéros

8	5	0	0	3
7	6	1	0	1
7	9	9	10	0
0	3	10	17	9

=Fin⇒

0	0	1	0	0
0	0	0	1	0
0	0	0	0	1
1	0	0	0	0
0	1	0	0	0

Une affectation de coût minimal est : $9 + 5 + 5 + 4 + 9 = 32$

2.12 Résolution du Problème de flot

Dans le problème de flot maximal, on cherche parmi les vecteurs de flot admissibles, ayant une divergence nulle en tout nœud différente de s ou t, celui qui maximise la divergence de s.

Nous présentons maintenant les algorithmes de flots, qui vont nous permettre de traduire de manière efficace le résultat théorique d'intégrité des flots. Nous commençons par le problème du flot de valeur maximale, car l'algorithme est plus simple à présenter dans ce cas. Nous cherchons donc un flot de valeur maximale v d'une source s à un puits p, qui soit admissible, c'est-à-dire qui vérifie la contrainte de capacité, puis nous présentons l'algorithme du flot maximum à coût minimum [3][7].

2.12.1 Algorithme de Ford Ful kerson

Présentation

l'algorithme de Ford Fulkerson permet de calculer un flot maximum entre un sommet source S et un sommet puit T.
le principe et de chercher à chaque itération une chaine améliorant μ ou joignant S et T, SI on

trouve une tell chaine alors on calcule l'augmentation du flot d'une valeur ε tel que

$$\varepsilon = \min\{\varepsilon_1, \varepsilon_2\}$$

avec

$$\varepsilon_1 = \min_{(i,j)\in\mu^+}\{\mu_{ij} - x_{ij}\}$$

$$\varepsilon_1 = \min_{(i,j)\in\mu^-}\{x_{ij}\}$$

en effet ,on peut augmenter la valeur du flot de ε unités sur les arcs avant de $\mu(\mu^+)$ et diminuer de ε unités sur les arc arrières de $\mu(\mu^-)$. l'algorithme se termine lorsque on arrive pas à trouver une chaine augmentant.

procédure du marquage

pour la recherche d'une chaine améliorante ,on utilisé la procédure de marquage suivant :

- Initialisation $S = \varnothing$, $\mu^+ = \mu^- = \varnothing$
- On marque le sommet s d'un " + " ,$S = \{s\}$.
- On marque d'une " + " une sommet $j \notin S$ tel que

$$x_{ij} < \mu_{ij}, \quad i \in S, \quad \mu^+ = \mu^+ \cup (i,j), \quad S = S \cup \{j\}$$

.

- On marque d'un " − " un sommet $j \notin S$, tell que

$$x_{ij} > 0, \quad i \in S, \quad \mu^- = \mu^- \cup (i,j), \quad S = S \cup \{j\}$$

.

- On arrête cette procédure lorsqu'on marque le sommet puit T .

Algorithme

1. Initialisation : $k = 0$, partir d'un flot initial réalisable ,par exemple $x = 0$.
2. 0 l'itération k, soit x^k un flot réalisable
   - trouver une chaine augmente μ^k reliant s et t en utilisant la procédure de marquage .
   - S'il n'en n'existe pas , alors x^k est un flot maximum ,arrêter l'algorithme sinon aller à (3).
3. Mise à jour du flot x^{k+1}

   soit ε^k la capacité réalisable de la chaine μ^k alors on pose

$$x_{ij}^{k+1} = \begin{cases} x_{ij}^k + \varepsilon^k & \text{si } (i,j) \in \mu^{k+}. \\ x_{ij}^k - \varepsilon^k & \text{si } (i,j) \in \mu^{k-}. \\ x_{ij}^k & \text{si } (i,j) \notin \mu^k. \end{cases}$$

   $k = k + 1$, aller à (2).

   {la coupe minimale s est composée de tous les sommets marqués durant la dernière itération de l'algorithme .}

Considèron le réseau $R = (X, U, c)$,ou $s = v_1$ et $p = v_7$

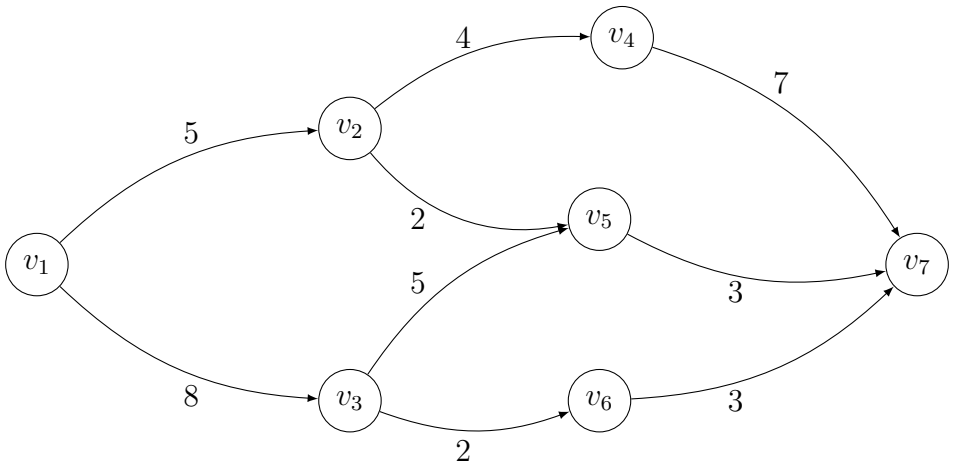


FIGURE 2.5 – Problème de flot maximum

Initialisation

$S = \varnothing$, $\mu^+ = \mu^- = \varnothing$

On marque le sommet s (entrée du réseau R) par le signe +, $S = \{s\}$

Itération 1

On pose : $\mu = \{v_1, v_2, v_4, v_7\}$

$\mu^+ = \{(v_1, v_2), (v_2, v_4), (v_4, v_7)\}$

$\mu^- = \varnothing$

$\varepsilon = \min\{(5-0), (4-0), (7-0)\} = \min\{5, 4, 8\} = 4$

On améliore ainsi le flot x^0 pour obtenir un nouveau flot $x^1 = 4$ Comme indiqué dans la figure 3.6 suivant :

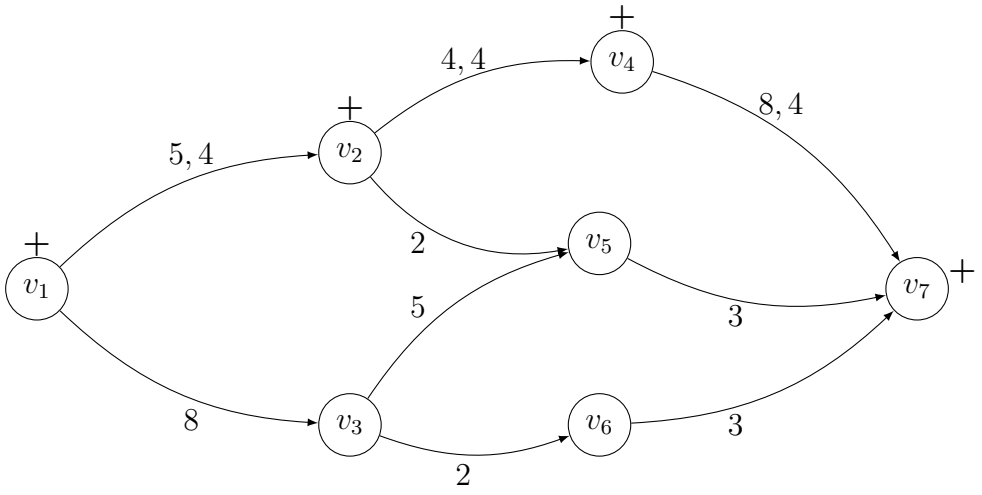


FIGURE 2.6 – Le réseau après avoir défini le nouveau flot x1

Itération 2

On pose : $\mu = \{v_1, v_2, , v_5, v_7\}$

$\mu^+ = \{(v_1, v_2), (v_2, v_5), (v_5, v_7)\}$

$\mu^- = \varnothing$

$\varepsilon = \min\{(5-4), (7-0), (2-0), (3-0)\} = \min\{1, 2, 3, \} = 1$

on améliore ainsi le flot x^1 pour obtenir un nouveau flot $x^2 = 4 + 1 = 5$, en ajoutant la quantité ε au flot des arcs de μ. Le flux des arcs n'appartenant pas à la chaine, reste inchangé.

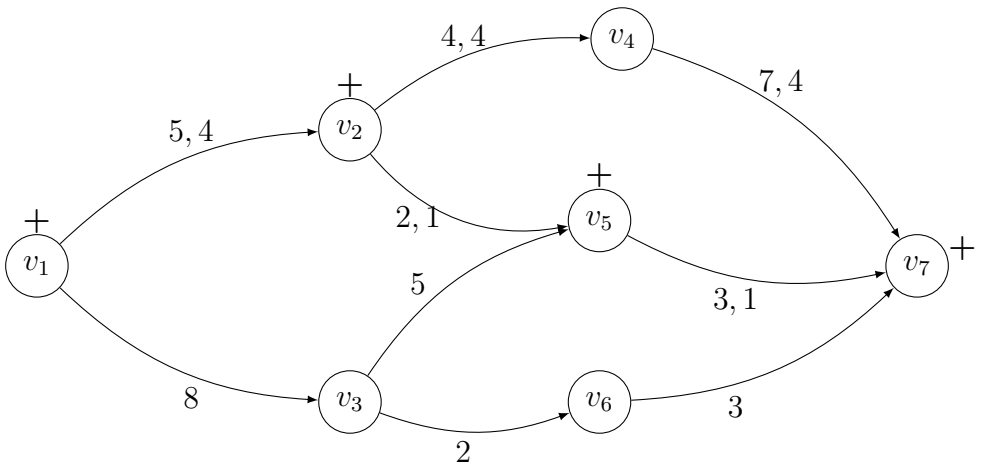


FIGURE 2.7 – Le réseau après avoir défini le nouveau flot x2

Itération 3

On pose : $\mu = \{v_1, v_3, v_5, v_7\}$

$\mu^+ = \{(v_1, v_3), (v_3, v_5), (v_5, v_7)\}$

$\varepsilon^+ = \min\{(8-0), (5-0), (3-1)\} = \min\{8, 5, 2\} = 2$

On améliore ainsi le flot x^2 pour obtenir un nouveau flot $x^3 = 5 + 2 = 7$, en ajoutant la quantité ε au flot des arcs de μ et retranchant la quantité μ au flot des arcs ε^-. Le flux des arcs n'appartenant pas à la chaine, reste inchangé.

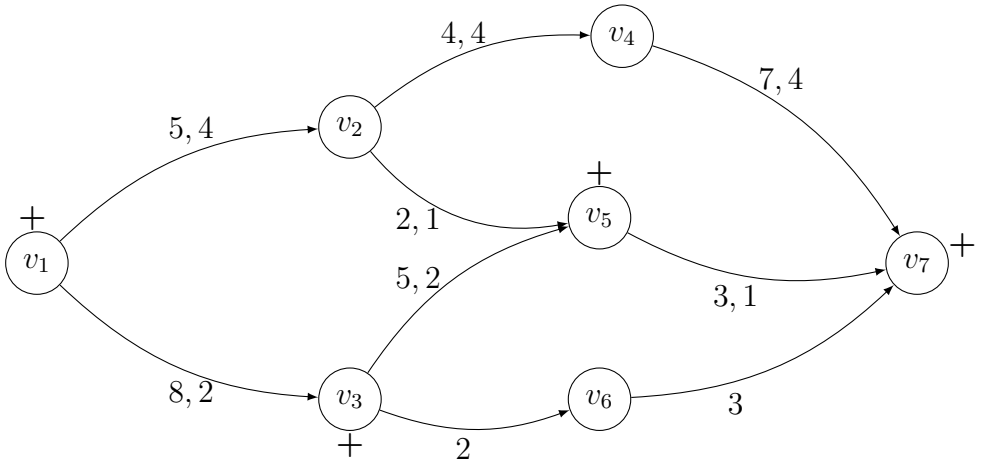


FIGURE 2.8 – Le réseau après avoir défini le nouveau flot x3

Itération 4

On pose : $\mu = \{v_1, v_3, v_6, v_7\}$

$\mu^+ = \{(v_1, v_3), (v_3, v_6), (v_6, v_7)\}$

$\varepsilon^+ = \min\{(8-2), (2-0), (3-0)\} = \min\{6, 2, 3\} = 2$

On améliore ainsi le flot x^3 pour obtenir un nouveau flot $x^4 = 5 + 2 + 2 = 9$, en ajoutant la quantité ε au flot des arcs de μ et retranchant la quantité μ au flot des arcs ε^-. Le flux des arcs n'appartenant pas à la chaine, reste inchangé.

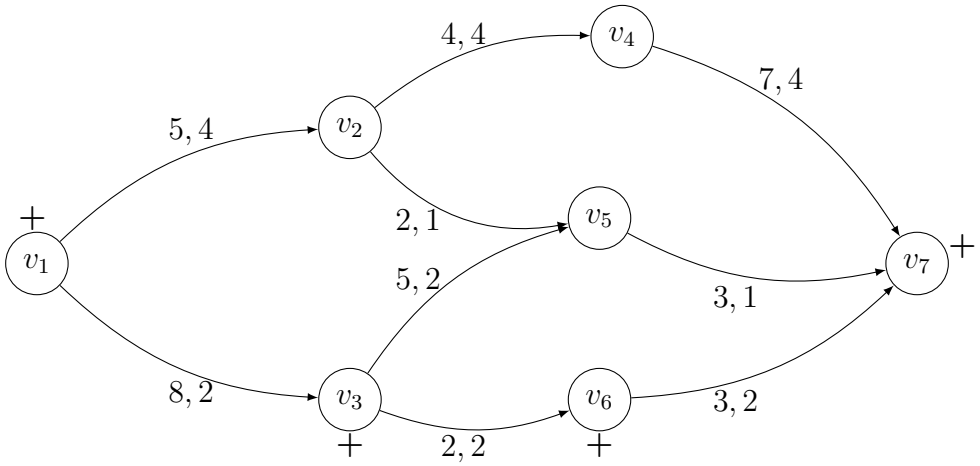


FIGURE 2.9 – Le réseau après avoir défini le nouveau flot x4

Itération 5 : la procédure de marquage s'arrête sans que le somment puits P ne sais marqué donc le flot de l'itération précédente est maximum $\varepsilon^+ = 9$

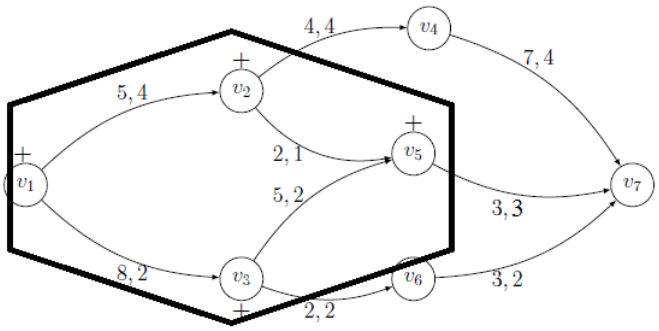


FIGURE 2.10 – Le réseau après avoir défini le flot maximum et la coupe minimal

la coupe minimale est forme des sommets marque d'un " + " ou d'un " − "

$S = \{v_1, v_2, v_3, v_4\}$ $\quad \Gamma(S) = \{(v_2, v_4), (v_5, v_7)?(v_3, v_6)\}$

donc $C(S) = 4 + 3 + 2 = 9$

CHAPITRE 3

RÉSOLUTION DE QUELQUE PROBLÈME DE TRANSPORT

Dans ce chapitre, nous allons aborder dans un premier temps le problème de transport avec les méthodes de résolution en suite le problème d'affectation avec l'algorithme de résolution hongrois, à la fin nous allons présenter le problème de flot et plus précisément le problème de flot maximum à coût minimal.

3.1 Résolution de problème de transport containeurs

3.1.1 Détermination d'une solution de base initiale :

Une compagnie Algérienne de transport doit envoyer de 3 dépôt (Constantine, Sétif, Alger) des conteneurs vides en destination de port(Alger, Bejaia ,Jijel,Annaba, Oran)

	Nr. de containeurs
Constantine	12
Setif	18
Alger	36

. Les besoins en containeurs dans les ports sont résumés ci-dessous :

	Besoins en containeurs
Alger	28
Béjaia	6
jijel	14
Annaba	4
Oran	14

Le transport des containeurs par des péniches. Chaque péniche ne peut contenir que un seul containeurs car les conteneur son de taille 40 et le coût du transport (par péniche) est proportionnel à la distance parcourue (130DA/km). Les distances sont données dans le tableau suivant :

	Alger	Bejaia	jijel	Annaba	Oran	offre
Constantine	388	230	132	149	785	12
Setif	269	107	133	272	667	18
Alger	18	180	331	573	414	36
Demande	28	6	14	4	14	66

TABLE 3.1 – Problème de transport initial

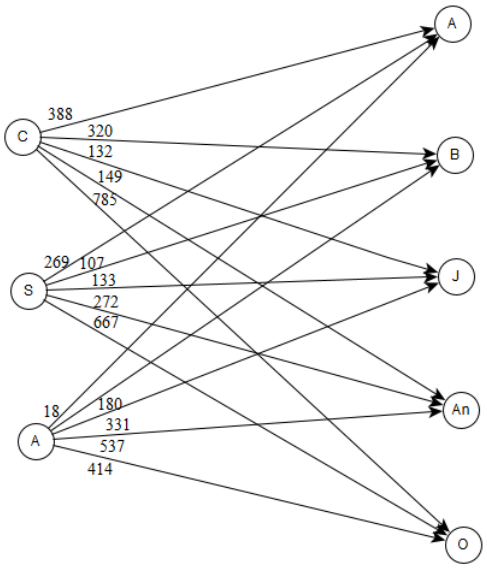


FIGURE 3.1 – le graphe assossis au problème de transport initial

Modéliser le problème en PL et le résoudre numériquement.

3.1.2 Formulation mathématique

Le modèle de transport deviendra alors :

$$
\begin{cases}
\min z = & 388x_{11} + 230x_{12} + 132x_{13} + 149x_{14} + 785x_{15} + 269x_{21} + 107x_{22} + 133x_{23} + 272x_{24} + 667x_{25} \\
& +18x_{31} + 180x_{32} + 331x_{33} + 573x_{34} + 414x_{35} + \\
\\
& x_{11} + x_{12} + x_{13} + x_{14} + x_{15} \leq 12 \\
& x_{21} + x_{22} + x_{23} + x_{24} + x_{25} \leq 18 \\
& x_{31} + x_{32} + x_{33} + x_{34} + x_{35} \leq 140 \\
\\
& x_{11} + x_{21} + x_{31} \leq 28 \\
& x_{12} + x_{22} + x_{32} \leq 6 \\
& x_{13} + x_{23} + x_{33} \leq 14 \\
& x_{14} + x_{24} + x_{34} \leq 4 \\
& x_{15} + x_{25} + x_{35} \leq 14 \\
\\
& \forall x_{ij} \geq 0
\end{cases}
$$

Dans la méthode du coin nord-ouest, la plus grande répartition possible est faite à la case dans le coin supérieur gauche du tableau, suivie d'allocations vers des cases adjacentes. Nous attribuons d'abord autant que possible à la case 1A (coin nord-ouest). Ce montant est de 12 Contenairs , puisqu'il s'agit du maximum pouvant être fourni par le dépôt de la ville constantine , Même si 28 containeurs sont demandées par le port. dans cette allocation initiale, est présentée dans le tableau 4.1. Nous allons ensuite allouer à une case adjacente à la case 1A, dans ce cas soit la case 2A où la case 1B. Cependant, la case 1B ne représente plus une répartition possible, car le le nombre de containeurs disponible à la ville Constantine a été alloué. Ainsi, la case 2A représente la seule alternative possible, et autant que possible est alloué à cette case, Le montant alloué à 2A peut être de 16 containeurs, la fourniture disponible à partir de la source 2 est 18 containeurs, le nombre actuellement demandé par la ville d'Alger est de 16 , Parce le nombre 16 est le plus contraint, il est attribué à la case 2A. Comme le montre le tableau 4.2. La troisième répartition est faite de la même manière que la deuxième répartition la seule case possible adjacente à la case 2A est la case 2B. Le plus qui peut être attribué soit de 2 containeurs (le nombre restent dans la ville de setif) la troisième allocation est de 4 containeurs a la case 3B est la quatrième est de 14 containeurs a la case 3c la cinquième allocation est la case 3D et elle est de 4 containeurs et enfin la dernier allocation est pour la case 3E est elle de 14 containeurs

Les allocations effectuées par la méthode de coin nord-ouest produisent donc une solution de base réalisable, puisque (m + n - 1) = 3 +5 - 1 = 7, ce qui équivaut au nombre d'allocations réalisées.

Comme le nombre de cases occupées 7 est égal (3 + 3-1), la condition est satisfaite, La solution initiale est complète lorsque toutes les exigences sont satisfaites.

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388 12	230	132	149	785	12
setif	269 16	107 2	133	272	667	18 2
Alger	18	180 4	331 14	573 4	414 14	36
Demande	28	6	14	4	14	66

TABLE 3.2 – Solution obtenue par la méthode de Coin Nord-Ouest

Le Cout Totale de transport est : $z = 388 \times 12 + 269 \times 16 + 107 \times 2 + 180 \times 4 + 331 \times 14 + 573 \times 4 + 414 \times 14$
$= 4656 + 4306 + 204 + 720 + 4634 + 2292 + 5796 = 22616 \times 130 = 2940080$

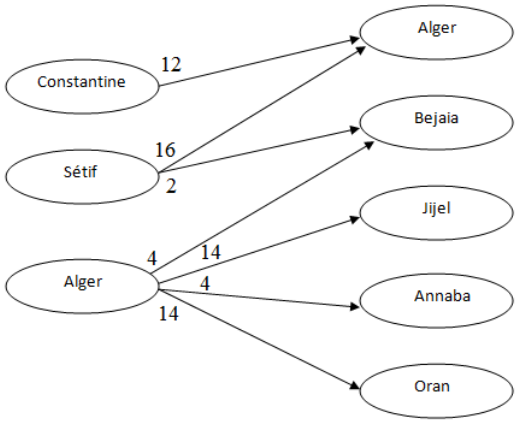


FIGURE 3.2 – Le graphe associé a la Table 4.2

2. Méthode Cout minimum

itération 1 :

Destinations / Origines	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	~~388~~ ~~28~~	230	132	149	785	12
setif	~~269~~ ~~28~~	107	133	272	667	18
Alger	~~18~~ 28	180	331	573	414	36 8
Demande	28	6	14	4	14	66

itération 2 :

Destinations / Origines	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	~~388~~ ~~28~~	~~230~~ ~~6~~	132	149	785	12
setif	~~269~~ ~~28~~	~~107~~ ~~6~~	133	272	667	18 12
Alger	~~18~~ 28	~~180~~ ~~6~~	331	573	414	36 8
Demande	28	6	14	4	14	66

itération 3 :

Destinations / Origines	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	/388/	230/	132/ 12	149/	785/	12 0
setif	269/	107/ 6	133	272	667	18 12
Alger	18/ 28	180/	331	573	414	36 8
Demande	28	6	14	4	14	66

itération 4 :

Destinations / Origines	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	/388	230/	132/ 12	149/	785/	12 0
setif	269/	107/ 6	133/ 2	272	667	18 10
Alger	18/ 28	180/	331/	573	414	36 8
Demande	28	6	14	4	14	66

itération 5 :

Destinations / Origines	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	/388	230/	132/ 12	149/	785/	12 0
setif	269/	107/ 6	133/ 2	272/ 4	667	18 6
Alger	18/ 28	180/	331/	573/	414	36 8
Demande	28	6	14	4	14	66

itération 6 :

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	/388	230/	132/ 12	149/	785/	120
setif	269/	107/ 6	133/ 2	272/ 4	667/ 6	180
Alger	18/ 28	180/	331/	573/	414	368
Demande	28	6	14	4	14	66

itération 7 :

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	/388	230/	132/ 12	149/	785/	120
setif	269/	107/ 6	133/ 2	272/ 4	667/ 6	180
Alger	18/ 28	180/	331/	573/	573/ 8	360
Demande	28	6	14	4	14	66

Le coût du transport est calculé en évaluant la fonction objective : $Z = 28 \times 18 + 6 \times 107 + 12 \times 32 + 133 \times 2 + 272 \times 4 + 667 \times 6 + 414 \times 8 = 504 + 642 + 266 + 1584 + 1088 + 4002 + 3312 = 11398 = 11398 \times 130 = 1481740$

Remarque :

- La solution de base initiale obtenue par la méthode de Minimum est de coût total égale à 1481740.

- La méthode de Minimum fournira une solution avec un coût inférieur à celui de la solution de coin nord-ouest car elle considère le coût de transport dans le processus d'allocation.

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388	230	132 12	149	785	12
setif	269	107 6	133 2	272 4	667 6	18
Alger	18 28	180	331	573	414 8	36
Demande	28	6	14	4	14	66

3.Méthode d'approximation de vogel

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

La premier affectation VAM :

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre	Différence ligne
Constantine	388	230	132	149	785	12	17
setif	269	107	133	272	667	18	26
Alger	18	180	331	573	414 14	36	162
Demande	28	6	14	4	14	66	
Différence colonne	251	73	1	123	253		

La deuxième affectation VAM :

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre	Différence ligne
Constantine	388	230	132	149	785	12	17
setif	269	107	133	272	667	18	26
Alger	18 22	180	331	573	414 14	36	162
Demande	28	6	14	4	14	66	
Différence colonne	251	73	1	123	253		

on répète la même démarche jusqu'à la dernière affectation

La dernière affectation VAM :

Destinations / Origines	Alger	Bejaia	Jijel	Annaba	Oran	offre	Différence ligne
Constantine	388 6	230	132 2	149 4	785	12	17
setif	269	107 6	133 12	272	667	18	26
Alger	18 22	180	331	573	414 14	36	162
Demande	28	6	14	4	14	66	
Différence colonne	251	73	1	123	253		

Coûts de pénalité récompensés après la dernière affectation.

Le Cout Totale de transport est : $z = 338 \times 6 + 132 \times 2 + 149 \times 4 + 107 \times 4 + 133 \times 12 + 18 \times 22 + 414 \times 14$
$= 2328 + 264 + 596 + 642 + 1596 + 396 + 5796 = 11618 \times 130 = 1510340$

3.1.3 Optimisation de solution de base initiale :

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

Méthode de Steeping-stone :

La méthode stepping-stone détermine s'il existe une cellule sans allocation qui réduirait le coût s'il est utilisé Le principe de la solution de base dans un problème de transport est de déterminer si une route de transport qui n'est pas actuellement utilisée (c'est-à-dire une cellule vide) entraînerait un coût total plus faible si elle était utilisée. Par exemple, le tableau 4.3 montre cinq cellules vides (A1, A2, B1, B3, C3, D1, D3, E1) représentant des itinéraires inutilisés

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388	230	132 12	149	785	12
setif	269	107 6	133 2	272 4	667 6	18
Alger	18 28	180	331	573	414 8	36
Demande	28	6	14	4	14	66

TABLE 3.3

la solution de base du Coût Minimum :

Notre première étape dans la méthode de stepping stone est d'évaluer ces cellules vides pour voir si l'utilisation de l'un d'entre eux réduirait le coût total. Si nous trouvons une telle route.

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388 +	230	132 -	149	785	12
setif	269	107	133 +	272	667 -	18
Alger	18 -	180	331	573	414 +	36
Demande	28	6	14	4	14	66

$A1 = 388 - 18 + 414 - 667 + 133 - 132 = 382$

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388	230	132	149	785	12
setif	269 +	107	133	272	667 -	18
Alger	18 -	180	331	573	414 +	36
Demande	28	6	14	4	14	66

$A2 = 269 - 18 + 414 - 667 = -302$

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388	230 +	132 -	149	785	12
setif	269	107 -	133 +	272	667	18
Alger	18	180	331	573	414	36
Demande	28	6	14	4	14	66

$B1 = 230 - 132 + 133 - 107 = 124$

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388	230	132	149	785	12
setif	269	107 -	133	272	667 +	18
Alger	18	180 +	331 +	573	414 -	36
Demande	28	6	14	4	14	66

$B3 = 180 - 107 + 667 - 414 = 326$

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388	230	132	149	785	12
setif	269	107	133 -	272	667 +	18
Alger	18	180	331 +	573	414 -	36
Demande	28	6	14	4	14	66

$C3 = 331 - 414 - 133 + 667 = 451$

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388	230	132 -	149 +	785	12
setif	269	107	133 +	272 -	667	18
Alger	18	180	331	573	414	36
Demande	28	6	14	4	14	66

$D1 = -132 + 149 - 272 + 133 = -122$

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388	230	132	149	785	12
setif	269	107	133	272 -	667 +	18
Alger	18	180	331	573 +	414 -	36
Demande	28	6	14	4	14	66

$D3 = -272 + 573 - 414 + 667 = 554$

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388	230	132 -	149	785 +	12
setif	269	107	133 +	272	667 -	18
Alger	18	180	331	573	414	36
Demande	28	6	14	4	14	66

$E1 = 785 - 667 + 133 - 132 = 119$

La case que on doit modifié est les cases A2 et D1 :

Modification de la case A2

Destinations / Origines	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388	230	132	149	785	12
setif	269 6	107 6	133 2	272 4	667 0	18
Alger	18 22	180	331	573	414 8	36
Demande	28	6	14	4	14	66

Modification de la case D1

Destinations / Origines	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388	230	132 8	149 4	785	12
setif	269	107	133 6	272 0	667	18
Alger	18	180	331	573	414	36
Demande	28	6	14	4	14	66

$A1 = 132 + 133 - 269 + 388 = 384$

$B1 = 230 - 132 + 133 - 107 = 124$

$B3 = 180 - 18 + 269 - 107 = 298$

Origines \ Destinations	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	388	230	132 8	149 4	785	12
setif	269 6	107 6	133 6	272	667	18
Alger	18 22	180	331	573	414 14	36
Demande	28	6	14	4	14	66

$C3 = 180 - 18 + 269 - 133 = 449$
$D2 = 272 - 133 + 132 - 149 = 122$
$D3 = 572 - 18 + 269 - 133 + 132 - 149 = 674$
$E1 = 785 - 132 + 133 - 269 + 18 - 414 = 121$
$E2 = 667 - 269 + 18 - 414 = 2$
$S = 18\times22+269\times6+107\times6+133\times6+132\times8+149\times4+414\times14 = 10898\times130 = 1416740$

3.1.4 Méthode de distribution modifiée :

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

Dans la table, la colonne supplémentaire à gauche avec les symboles U_i et la ligne supérieure supplémentaire avec les symboles V_j représentent des valeurs qui doivent être calculées. Calculer pour toutes les cellules allouées : $U_i + V_j = C_{ij}$: Coût de transport unitaire pour la case ij. Formules pour les cellules contenant des allocations :
$X_{1c} : U_0 + V_2 = 132$
$X_{2B} : U_1 + V_1 = 107$
$X_{2C} : U_1 + V_2 = 133$
$X_{2D} : U_1 + V_3 = 272$
$X_{2E} : U_1 + V_4 = 667$
$X_{3A} : U_2 + V_0 = 28$
$X_{3E} : U_2 + V_4 = 8$

Maintenant, il existe 7 équations avec 8 inconnues. Pour résoudre ces équations, il est nécessaire d'assigner une seule des inconnues à une valeur nulle. il s'agit de mettre
$U_0 = 0\,; V_2 = 132\,; U_1 = -1\,; V_1 = 108\,; V_3 = 273\,; V_4 = 668\,; U_2 = -660\,; V_0 = 668.$

Destinations / Origines	U_i	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	U_0 0	388	230	132 12	149	785	12
setif	U_1 -1	269	107 6	133 2	272 4	667 6	18
Alger	U_2 -660	18 28	180	331	573	414 14	36
V		V_0 688	V_1 108	V_2 132	V_3 275	4 668	V_5
Demande		28	6	14	4	14	

$S = 132\times12+269\times6+107\times6+133\times2+272\times4+18\times22+414\times14 = 11386\times130 = 1480180$

Changement de cout pour les cellules vides :
$C_{ij} - U_{ij} - V_{ij} = \Delta_{ij}$

$X_{1A} = \Delta_{1A} = C_{1A} - U_0 - V_0 = 388 - 0 - 688 = -300$
$X_{2A} = \Delta_{2A} = C_{2A} - U_1 - V_0 = 269 + 1 - 688 = -418$
$X_{1B} = \Delta_{1B} = C_{1B} - U_0 - V_0 = 388 - 0 - 688 = 50$
$X_{3B} = \Delta_{3B} = C_{3B} - U_2 - V_1 = 180 + 660 - 108 = 708$
$X_{3C} = \Delta_{3C} = C_{3C} - U_2 - V_2 = 331 + 660 - 132 = 1123$
$X_{1D} = \Delta_{1D} = C_{1D} - U_0 - V_3 = 149 - 0 - 275 = -126$
$X_{3D} = \Delta_{3D} = C_{3D} - U_2 - V_3 = 573 + 660 - 275 = 1508$
$X_{1E} = \Delta_{1E} = C_{1E} - U_0 - V_4 = 785 - 0 - 668 = 117$

$U_0 + V_2 = 132$
$U_1 + V_0 = 269$
$U_1 + V_1 = 107$
$U_1 + V_2 = 133$
$U_1 + V_3 = 272$
$U_2 + V_0 = 18$

Origines \ Destinations	U_i	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	U_0 0	388	230	132 12	149	785	12
setif	U_1 1	269 6	107 6	133 2	272 4	667	18
Alger	U_2 -250	18 22	180	331	573	414 14	36
V		V_0 268	V_1 106	V_2 132	V_3 271	$_4$ 664	V_5
Demande		28	6	14	4	14	

$U_2 + V_4 = 414$

$U_0 = 0\,; V_2 = 132\,; U_1 = 1\,; V_1 = 106\,; V_3 = 271\,; V_4 = 664\,; U_2 = -250\,; V_0 = 268.$

$X_{1A} = \Delta_{1A} = C_{1A} - U_0 - V_0 = 388 - 0 - 268 = 120$
$X_{1B} = \Delta_{1B} = C_{1B} - U_0 - V_1 = 230 - 0 - 106 = 124$
$X_{3B} = \Delta_{3B} = C_{3B} - U_2 - V_1 = 180 + 250 - 106 = 324$
$X_{3C} = \Delta_{3C} = C_{3C} - U_2 - V_2 = 331 - 250 - 132 = 449$
$X_{1D} = \Delta_{1D} = C_{1D} - U_0 - V_3 = 149 - 0 - 271 = -122$
$X_{3D} = \Delta_{3D} = C_{3D} - U_2 - V_3 = 573 + 250 - 271 = 552$
$X_{1E} = \Delta_{1E} = C_{1E} - U_0 - V_4 = 785 - 0 - 664 = 121$
$X_{2E} = \Delta_{2E} = C_{2E} - U_1 - V_4 = 667 - 1 - 664 = 2$

On change la case 1D

Origines \ Destinations	U_i	Alger	Bejaia	Jijel	Annaba	Oran	offre
Constantine	U_0 0	388	230	132 8	149 4	785	12
setif	U_1 -16	269 6	107 6	133 6	272	667	18
Alger	U_2 -267	18 22	180	331	573	414 14	36
V		V_0 285	V_1 123	V_2 132	V_3 149	$_4$ 681	V_5
Demande		28	6	14	4	14	

$U_0 + V_2 = 132$
$U_0 + V_3 = 149$
$U_1 + V_0 = 269$
$U_1 + V_2 = 133$
$U_1 + V_1 = 107$
$U_1 + V_2 = 133$
$U_2 + V_0 = 18$
$U_2 + V_4 = 414$

$U_0 = 0\,; V_2 = 132\,; U_1 = -16\,; V_1 = 123\,; V_3 = 149\,; V_4 = 681\,; U_2 = -267\,; V_0 = 285.$

Changement de cout pour les cellules vides :

$X_{1A} = \Delta_{1A} = C_{1A} - U_0 - V_0 = 388 - 285 - 0 = 103$
$X_{1B} = \Delta_{1B} = C_{1B} - U_0 - V_1 = 230 - 0 - 123 = 107$
$X_{3B} = \Delta_{3B} = C_{3B} - U_2 - V_1 = 180 + 267 - 123 = 324$
$X_{3C} = \Delta_{3C} = C_{3C} - U_2 - V_2 = 331 + 267 - 132 = 466$
$X_{2D} = \Delta_{2D} = C_{2D} - U_1 - V_3 = 272 - 149 + 16 = -139$
$X_{3D} = \Delta_{3D} = C_{3D} - U_2 - V_3 = 573 + 267 - 149 = 157$
$X_{1E} = \Delta_{1E} = C_{1E} - U_0 - V_4 = 785 - 0 - 681 = 104$
$X_{2E} = \Delta_{2E} = C_{2E} - U_1 - V_4 = 667 + 16 - 681 = 2$

3.2 Application de l'algorithme Hongrois

3.2.1 Problème 2

considérons le problème suivant :
Une compagnie de taxis a un surplus d'une voiture dans les villes A, B, C et D mais il lui manque une voiture dans chacune des villes E, F, G et H. La matrice suivante donne en milles la distance entre les villes concernées :

	E	F	G	H
A	41	72	39	52
B	22	29	49	65
C	27	39	60	51
D	45	50	48	52

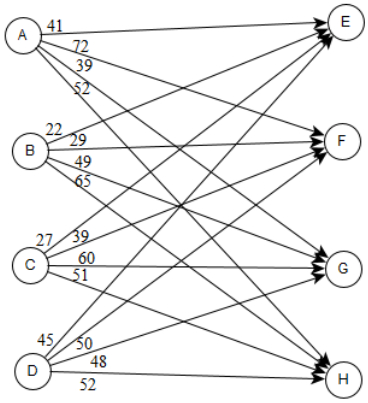


FIGURE 3.3 – le graphe associé au problème 2

3.2.2 Formulation mathématique

Le modèle d'affectation deviendra alors :

$$\begin{cases} \min z = & 41x_{11} + 72x_{12} + 39x_{13} + 52x_{14} + 22x_{21} + 29x_{22} + 49x_{23} + 65x_{24} \\ & +27x_{31} + 39x_{32} + 60x_{33} + 51x_{34} + 45x_{41} + 50x_{42} + 48x_{43} + 52x_{44} \\ & \\ & x_{11} + x_{12} + x_{13} + x_{14} = 1 \\ & x_{21} + x_{22} + x_{23} + x_{24} = 1 \\ & x_{31} + x_{32} + x_{33} + x_{34} = 1 \\ & x_{41} + x_{42} + x_{43} + x_{44} = 1 \\ & \\ & x_{11} + x_{21} + x_{31} + x_{41} = 1 \\ & x_{12} + x_{22} + x_{32} + x_{42} = 1 \\ & x_{13} + x_{23} + x_{33} + x_{43} = 1 \\ & x_{14} + x_{24} + x_{34} + x_{44} = 1 \\ & x_{15} + x_{25} + x_{35} + x_{45} = 1 \\ & \\ & \forall x_{ij} \geq 0 \end{cases}$$

itération 1

Réduction des ligne, on crée une nouvelle matrice des couts en choisissant le cout minimal sur chaque ligne et on le soustrayant de chaque cout sur la ligne

	E	F	G	H	min
A	41	72	39	52	39
B	22	29	49	65	22
C	27	39	60	51	27
D	45	50	48	52	45

=Réduction des lignes⇒

	E	F	G	H
A	2	33	0	13
B	0	7	27	43
C	0	12	33	24
D	0	5	3	7

exemple : pour la deuxième ligne (B) -Relation (B,E) :$22 - 22 = 0$

-Relation (B,F) :$29 - 22 = 7$

-Relation (B,G) :$49 - 22 = 27$

-Relation (B,H) :$65 - 22 = 43$

parfois, il peut arriver que la matrice a ce stade ne pas être utiliser pour utiliser pour attribuer cella est cas pour la matrice ci dessus dans le cas ci dessus aucune affectation peut faite. Noter que la tache E est faite efficacement a la fois par l'agent B et l'agent C et l'agent D. les trois ne peuvent pas être affectées de la même tache.
pour Y remédier nous répétons la procédure ci-dessus pour tout les colonne.

itération 2 :

Réduction des colonne : on crée une nouvelle matrice des couts en choisissant le cout minimal dans chaque colonne et en le soustrayant de chaque cout dans la colonne.

	E	F	G	H
A	2	28	0	13
B	0	2	27	43
C	0	7	33	24
D	0	0	3	7
min	0	5	0	7

itération 3 :
tout les zéros dans la matrice doivent être couvert parle marquage comme quelques ligne et/ou colonnes possible. la procédure suivante est une façon d'y arriver tout d'abord attribuer autant de tache que possible.
on cherche la ligne comportant le moin de zéros non barrés, en cas d'égalité on prendre la ligne la plus haute :

1. on encadre un des zéros de cette ligne /arbitrairement celui le plus a gauche
2. on barre tout les zéros se trouve sur la même ligne et colonne que celui sélectionné
3. on recommence jusqu'à ce que tous les zéros sont encadrés ou barrés

	E	F	G	H
A	2	28	$\boxed{0}$	6
B	$\boxed{0}$	2	27	36
C	$\not{0}$	7	33	17
D	$\not{0}$	$\boxed{0}$	3	$\not{0}$

étape 4 :

Le sous tableau permet par retranchement de posséder une configuration permettant de trouver une solution optimal. la construction des pivot se fait aussi :

1. on marque tout ligne ne contenant aucun zéro encadré
2. on marque toute colonne ayant un zéro barré sur une ligne marqué
3. on marque tout ligne ayant un zéro encadré dans une colonne marqué
4. Répéter 2 et 3 jusqu'à ne plus avoir de modification.

		X			
	E	F	G	H	
A	2	28	[0]	6	
B	[0]	2	27	36	X
C	~~0~~	7	33	17	X
D	~~0~~	[0]	3	~~0~~	

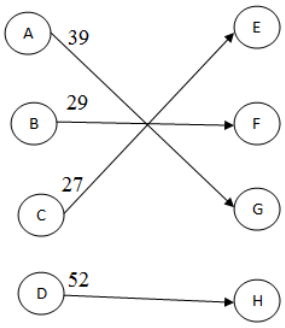
on trace alors un trait sur toute ligne non marquée et sur colonne marquée. Dans notre exemple la ligne 2 et 3 sont marquée,aussi que la colonne 4.

Étape 5 :
les cases non barrées par un trait constituant partiel

1. tranche a tout les cases de ce tableau le plus petit élément de celui-ci.
2. on ajoute cet élément a tout les case du tableau barrées dans les deux sens.

	E	F	G	H
A	4	28	[0]	6
B	~~0~~	[0]	25	34
C	[0]	5	31	15
D	2	~~0~~	3	[0]

on a le nombre minimal de ligne est égal a 4, nous obtenons un résultat optimal après la 1 er itération.
l'affectation minimal se calcul a partir du tableau initial : $39 + 29 + 27 + 52 = 147$



3.3 Application de l'algorithme de Ford ful kerson

3.3.1 problème d'alimentation de 4 villes par 3 châteaux d'eau :

trois châteaux d'eau alimentent 4 villes à travers un réseau de canalisations au sein duquel sein trouvent également une stations de pompage. Les châteaux d'eau ont une capacité limitée qui s'élève pour chacun d'eux à 62 000,75 000, 90 000 m3. Les villes ont exprimé une demande qui est au minimum de 48 000 pour la ville 1, 50 000 pour la 2 et 30 000 pour la ville 3, pour la ville 4 est de 490 000 en m3. Les canalisations entre les châteaux d'eau et les villes ont des débits limités. Par exemple, pour la canalisation reliant le château 1 à la ville 1, le débit maximum est de 36 000 alors que celui de la canalisation reliant la station de pompage à la ville 2 est de 50 en milliers de m3.

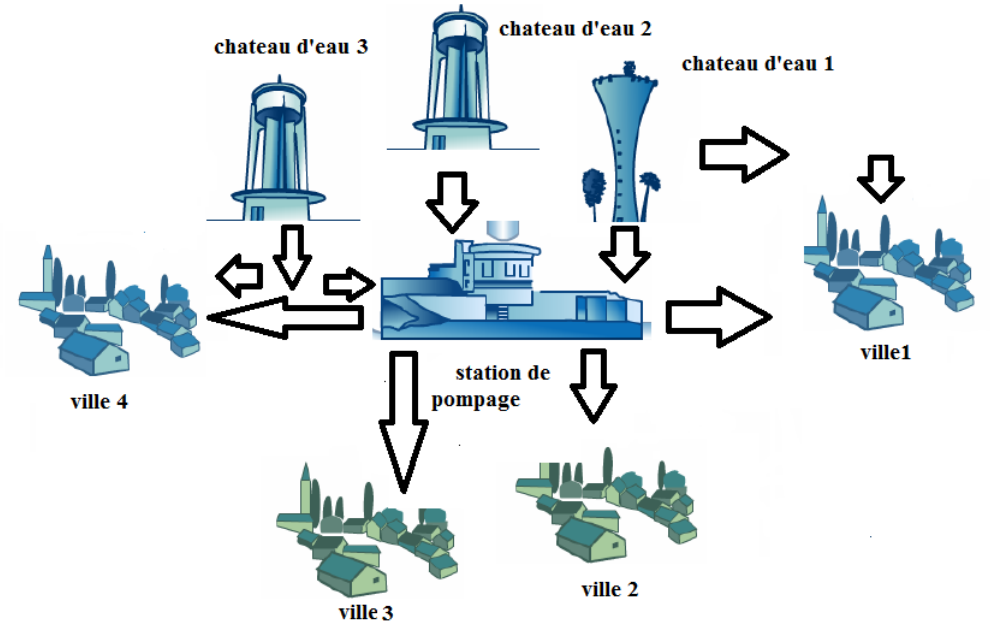


FIGURE 3.4

pour résoudre ce problème ont lui appliqué l'algorithme de Ford ful kerson on commence par le flot nul $F^0 = (0,0,0,0,0,0,0,0,0,0)$ de valeur $f^0 = 0$, et les sommets $S, C^1, C_2, C_3, P, V_1, V_2, V_3, V_4, ST$ représente les porte $0,1,2,3,4,5,6,7,8,9$

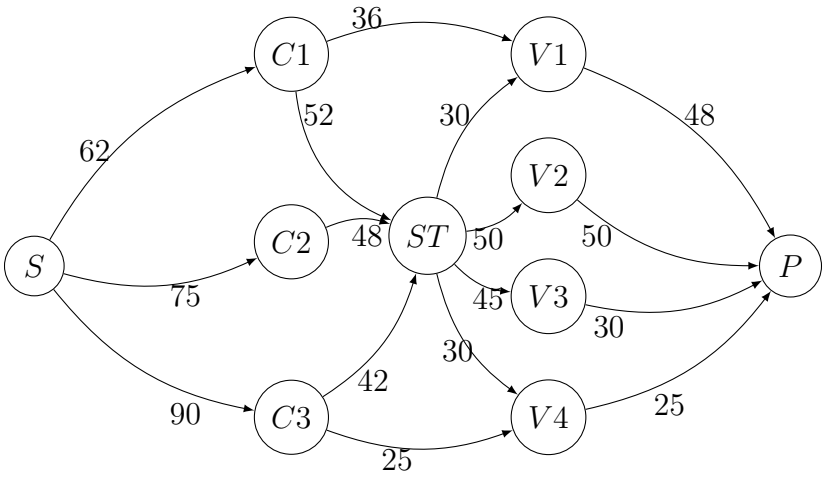


FIGURE 3.5 – problème de flot initial

Initialisation : on marque le sommet S par le signe + on pose $S = \{S\}$; $CH^+ \cup CH^- = \varnothing$ et on a $f^k = 0$ avec $k = 0$

Itération 1 : Dans le réseau R, on suit la procédure de marquage suivant : - on marque le sommet 1 d'un +, car il est successeur de S et $f(S,1) = 0 \leq c(s,1) = 62$
on pose : $CH^+ \cup [s,1]$, $s = S \cup \{1\}$

- on marque le sommet V1 d'un +, car il est successeur de 1 et $f(2,5) = 0 \leq c(2,5) = 36$

on pose : $CH^+ \cup [2,5], s = S \cup \{5\}$ - on marque le sommet P d'un +, car il est successeur de 5 et $f(5,P) = 0 \leq c(5,p) = 48$ on pose : $CH^+ \cup [5,p], s = S \cup \{p\}$ le sommet p étais marqué la procédure s'arrête, on obtient donc la chaine augmentant :
$CH = CH^+ \cup CH^- = CH_+ = [s,1,5,p]$ reliant le sommet S et P, on calcule :
$\varepsilon_1 = min\{c(u_k) - f(u_k); u_k \in CH^+\}$ = min {(s,1)-f(s,1) ;c(1,5)-f(1,5) : c(5,p)- f(5,p)} = min (62-0 ;36-0 ;48-0)=36 on améliore aussi le flot F^0 pour obtenir un nouveau flot F^1, on ajoutant la quantité $\varepsilon = 36$ au flot des arcs de CH^+ le flux des arcs n'appartenant pas à la chaine. reste inchangé et $f^1 = 0 + 6$

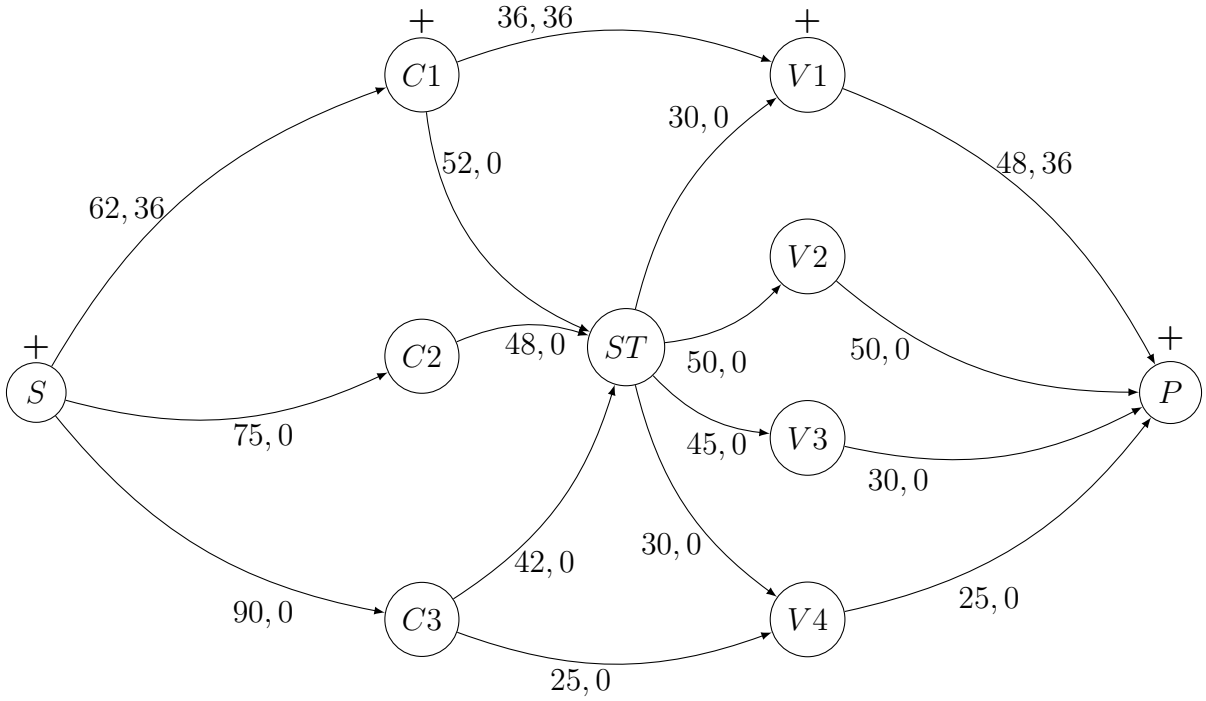


FIGURE 3.6 – Le réseau après avoir défini le nouveau flot

Itération 2 : Dans le Réseau, on suit la procédure de marquage suivante : en efface les marquage sauf en :

- on marque le sommet 3 d'un +, car il est successeur de S et $f(S,3) = 0 \leq c(s,3) = 90$ on pose : $CH^+ \cup [s,3]$, $s = S \cup \{3\}$

- on marque le sommet V4 d'un +, car il est successeur de C3 et $f(3,8) = 0 \leq c(3,8) = 25$

on pose : $CH^+ \cup [3,8], s = S \cup \{8\}$ - on marque le sommet P d'un +, car il est successeur de 8 et $f(8,P) = 0 \leq c(8,p) = 50$ on pose : $CH^+ \cup [5,p], s = S \cup \{p\}$ le sommet p étais marqué la procédure s'arrête, on obtient donc la chaine augmentant :
$CH = CH^+ \cup CH^- = CH_+ = [s,3,8,p]$ reliant le sommet S et P, on calcule :
$\varepsilon_1 = min\{c(u_k) - f(u_k); u_k \in CH^+\} = min\{(s,3) - f(s,3); c(3,8) - f(3,8) : c(8,p) - f(8,p)\} = min(90;25;50) = 25$ on améliore aussi le flot F^1 pour obtenir un nouveau flot F^2, on ajoutant la quantité $\varepsilon = 25$ au flot des arcs de CH^+ le flux des arcs n'appartenant pas à la chaine. reste inchangé et $f^2 = 25 + 36 = 61$

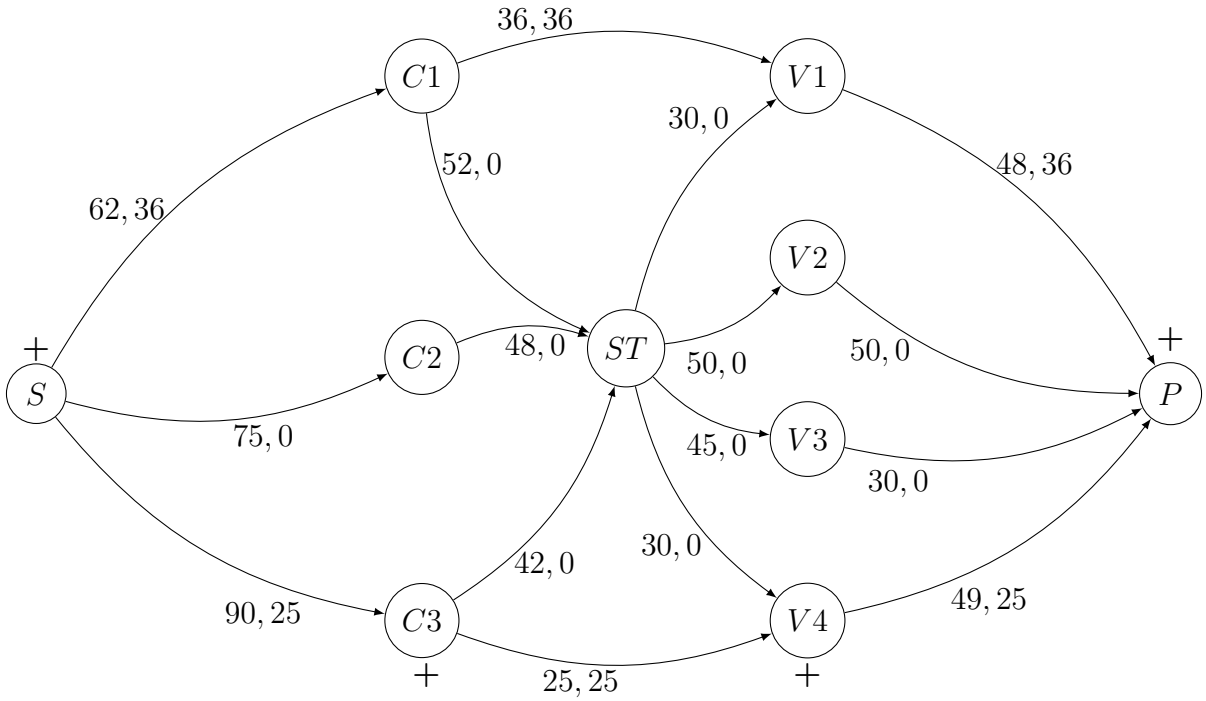


FIGURE 3.7 – Le réseau après avoir défini le nouveau flot

en suit le même raisonnement jusqu'à l'itération 7

Itération 7 : Dans le Réseau, on suit la procédure de marquage suivante : en efface les marquage sauf en S :

- on marque le sommet 3 d'un +, car il est successeur de S et $f(S,3) = 25 \leq c(s,3) = 90$ on pose : $CH^+ \cup [s,3]$, $s = S \cup \{3\}$

- on marque le sommet V4 d'un +, car il est successeur de C3 et $f(3,4) = 0 \leq c(3,4) = 42$

on pose : $CH^+ \cup [3,4], s = S \cup \{4\}$ - on marque le sommet 7 d'un +, car il est successeur de 4 et $f(4,7) = 12 \leq c(4,7) = 45$ on pose : $CH^+ \cup [4,7], s = S \cup \{7\}$ - on marque le sommet P d'un +, car il est successeur de 7 et $f(7,P) = 12 \leq c(7,p) = 30$ on pose : $CH^+ \cup [7,P], s = S \cup \{P\}$

le sommet p étais marqué la procédure s'arrête, on obtient donc la chaine augmentant : $CH = CH^+ \cup CH^- = CH_+ = [s,3,4,7,p]$ reliant le sommet S et P, on calcule : $\varepsilon_1 = min\{c(u_k) - f(u_k); u_k \in CH^+\} = min\{(s,3) - f(s,3); c(3,4) - f(3,4); c(4,7) - f(4,7); c(7,p) - f(7,p)\} = min(90-25; 42-0; 45-12; 30-12) = 18$ on améliore aussi le flot F^6 pour obtenir un nouveau flot F^7, on ajoutant la quantité $\varepsilon = 18$ au flot des arcs de CH^+ le flux des arcs n'appartenant pas à la chaine. reste inchangé et $f^8 = 135 + 18 = 153$

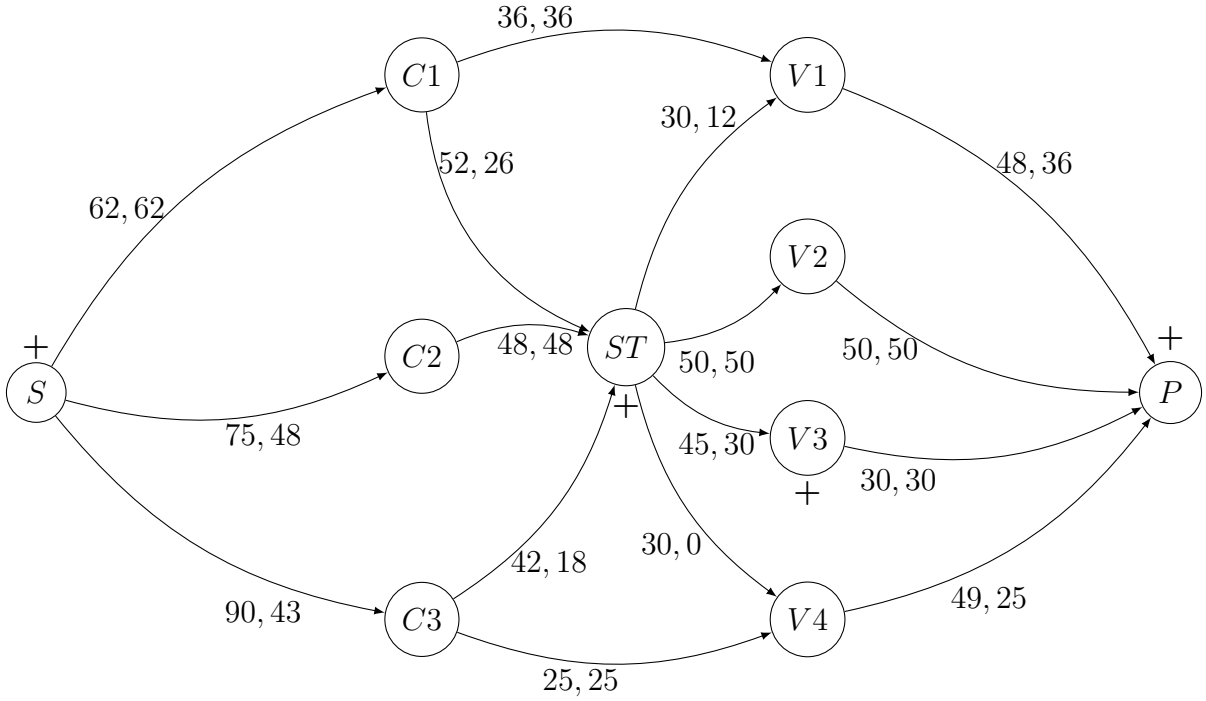


FIGURE 3.8 – Le réseau après avoir défini le nouveau flot

Itération 8 : Dans le Réseau, on suit la procédure de marquage suivante : en efface les marquage sauf en S :

- on marque le sommet 3 d'un +, car il est successeur de S et $f(S,3) = 25 \leq c(s,3) = 90$ on pose : $CH^+ \cup [s,3]$, $s = S \cup \{3\}$

- on marque le sommet V4 d'un +, car il est successeur de C3 et $f(3,4) = 0 \leq c(3,4) = 42$

on pose : $CH^+ \cup [3,4], s = S \cup \{4\}$ - on marque le sommet 8 d'un +, car il est successeur de 4 et $f(4,8) = 0 \leq c(4,7) = 3$ on pose : $CH^+ \cup [4,8], s = S \cup \{8\}$ - on marque le sommet P d'un +, car il est successeur de 8 et $f(8,P) = 25 \leq c(8,p) = 49$ on pose : $CH^+ \cup [8,P], s = S \cup \{P\}$

le sommet p étais marqué la procédure s'arrête, on obtient donc la chaine augmentant : $CH = CH^+ \cup CH^- = CH_+ = [s,3,4,8,p]$ reliant le sommet S et P, on calcule :
$\varepsilon_1 = min\{c(u_k) - f(u_k); u_k \in CH^+\} = min\{(s,3) - f(s,3); c(3,4) - f(3,4); c(4,7) - f(4,8); c(8,p) - f(7,p)\} = min(90-43; 42-18; 30-0; 49-25) = 24$ on améliore aussi le flot F^7 pour obtenir un nouveau flot F^8, on ajoutant la quantité $\varepsilon = 24$ au flot des arcs de CH^+ le flux des arcs n'appartenant pas à la chaine. reste inchangé et $f^8 = 153 + 24 = 177$
Donc $F^8 = F$ est un flot maximum .

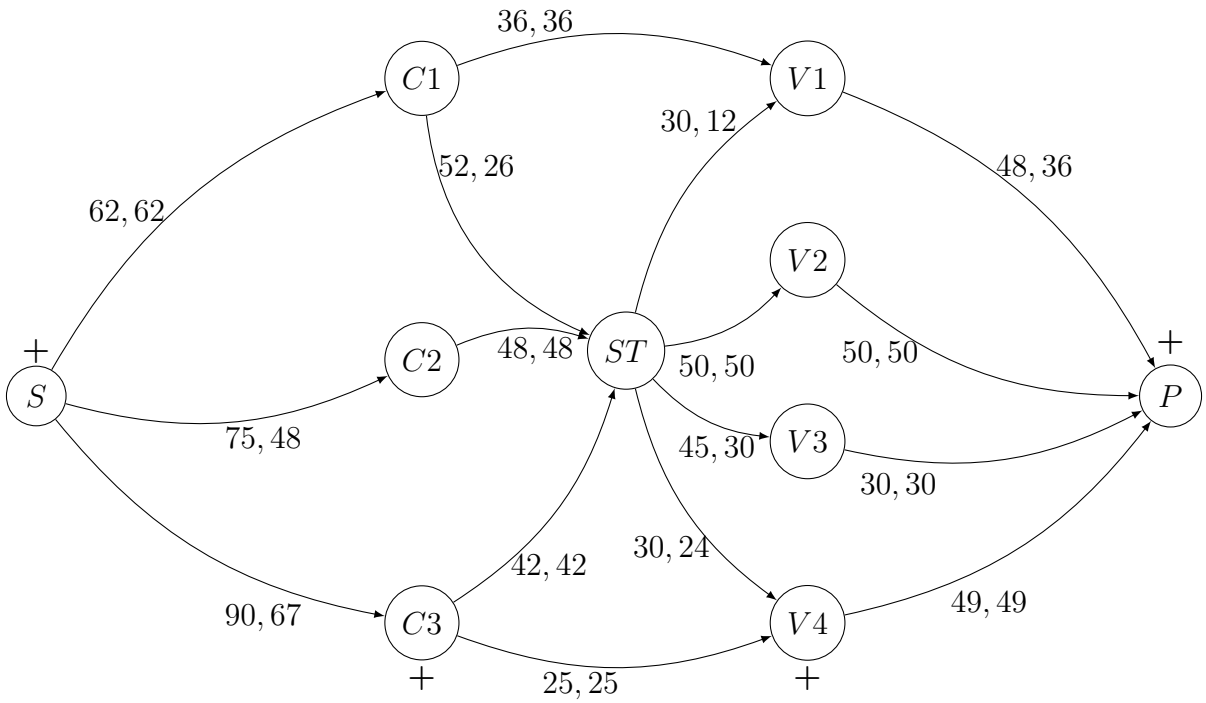


FIGURE 3.9 – Le réseau après avoir défini le nouveau flot maximum

la coupe minimale est forme des sommets marque d'un " + " $S = \{v_1, v_2, v_3, v_4\}$ $\Gamma(S) = \{(s,1),(2,4),(3,4),(3,8)\}$
donc $C(S) = 62 + 48 + 42 + 25 = 177$

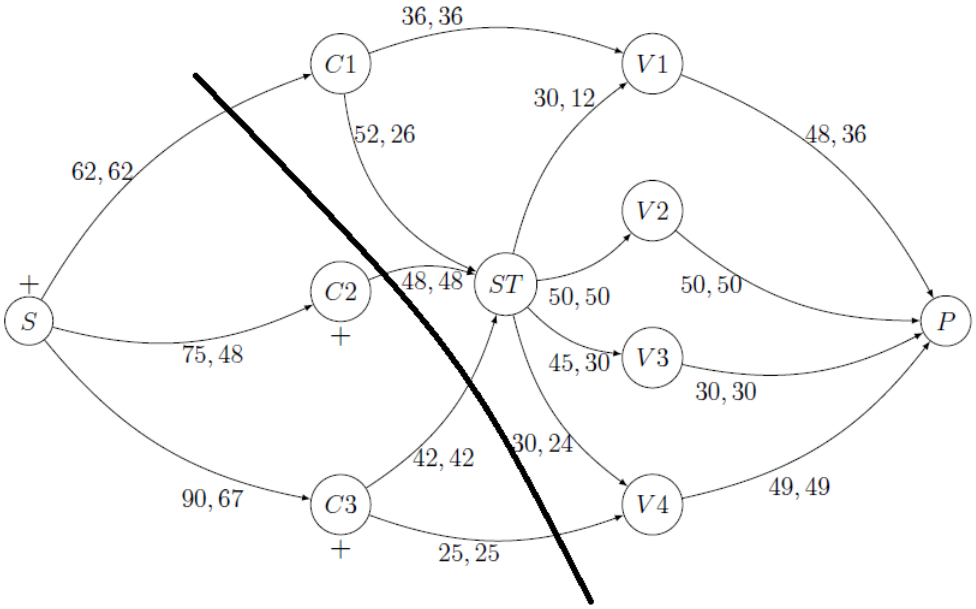


FIGURE 3.10 – coupe minimal

Le problème de transport est un problème qui peux être représenté sous forme d'un graphe et qu'on peux le résoudre en utilisant les différents méthodes de résolution.

CHAPITRE 4

APPLICATION ET ÉVALUATION

Le modèle de transport est un modèle linéaire, qui peut être traité par l'algorithme du simplexe. Cependant, la structure particulière de ce modèle permet de simplifier considérablement l'algorithme. Nous illustrons ici ce que devient l'algorithme lorsqu'on utilise les méthodes de la recherche de solution optimale de problème de transport présentées dans le chapitre précédent, en les appliquant pour faire face à quelques situations réelles afin d'expliquer leurs fonctionnements, et manipulant l'algorithme sous forme d'un programme informatique (programmé en langage C) capable de résoudre un problème de transport équilibré.

4.1 Langages utilisés

C est un langage de programmation impératif et généraliste. Inventé au début des années 1970 pour réécrire UNIX, C est devenu un des langages les plus utilisés. De nombreux langages plus modernes comme $C++$, Java et PHP reprennent des aspects de C. Le C est un langage de programmation de bas niveau très populaire, crée dans les années 1970 par D.Ritchie et B.W.Kernighan. Il est portable, libre, faiblement typé (peu de types de variables différents : son fonctionnement est donc proche de l'ordinateur (gain en rapidité), mais plus difficile à manipuler pour le programmeur). Le C n'est sans doute pas le langage le plus facile à apprendre (notamment à cause de l'adoption du concept parfois un peu obscure des pointeurs), ni le plus récent, mais ses qualités font de lui un langage incontournable en matière de programmation.

Le fameux Hello Word :

```
#include <stdio.h>
int main()
{
  printf("hello, world\n");
  return 0;
}
```

4.2 Technologies utilisées

Dev-C++ est un environnement de développement intégré (IDE) permettant de programmer en C et en C++. Développé avec Borland Delphi 6, Dev-C++ était disponible uniquement sous Microsoft Windows. Longtemps à l'abandon, le projet a été repris par un autre développeur en 2011 et est régulièrement mis à jour.

Cet IDE complet comprend entre autres un « répertoire de classes », servant à localiser facilement les fonctions, classes et membres du code source, un « répertoire de fonctions incluses », fonctionnant comme le répertoire de classes mais pour chercher dans les fichiers inclus (header), et un débogueur qui permet de surveiller l'état des variables pendant l'exécution du programme. Il souffre en revanche de l'absence d'un éditeur de ressources, ce qui rend la conception d'applications délicate si on ne fait pas appel à un outil externe.

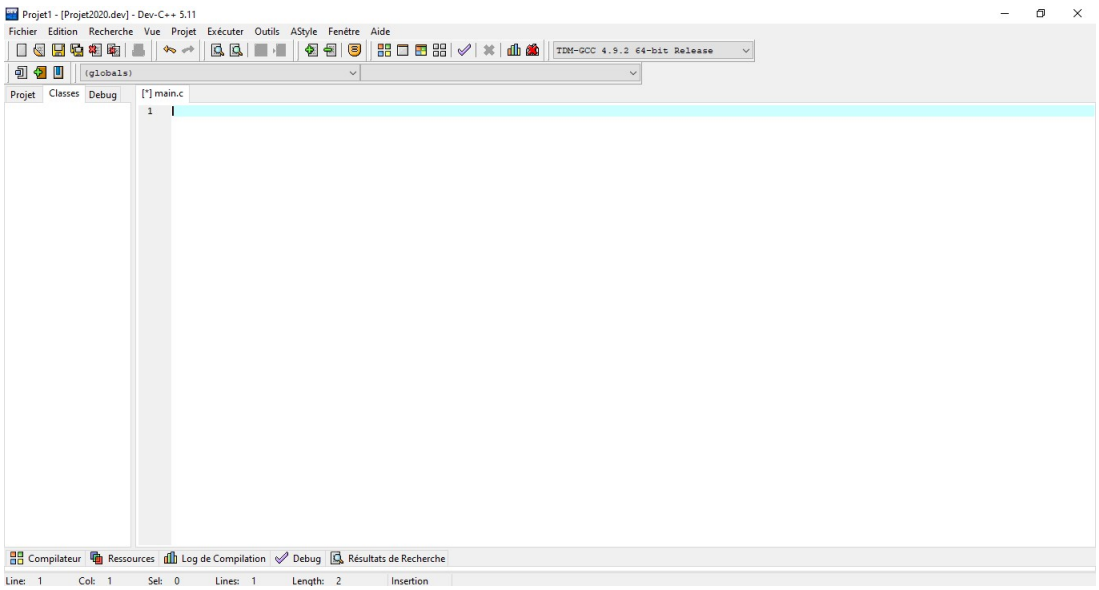


FIGURE 4.1 – Interface de Dev-C++

4.2.1 Fichier nouveau

Si votre programme tient dans un seul fichier et n'a pas besoin de bibliothèques particulières, vous pouvez utiliser Dev-C++ sans créer de projet. Pour cela il vous suffit de lancer Dev-C++ puis de créer un fichier source : commande **Nouveau > Fichier Source** du menu **Fichier** (beaucoup de commandes des menus s'obtiennent aussi par des boutons de la barre d'outils et/ou par des raccourcis clavier).

Enregistrez immédiatement ce fichier à l'aide des commandes **Sauvegarder** ou **Sauvegarder Sous...** du menu **Fichier**.

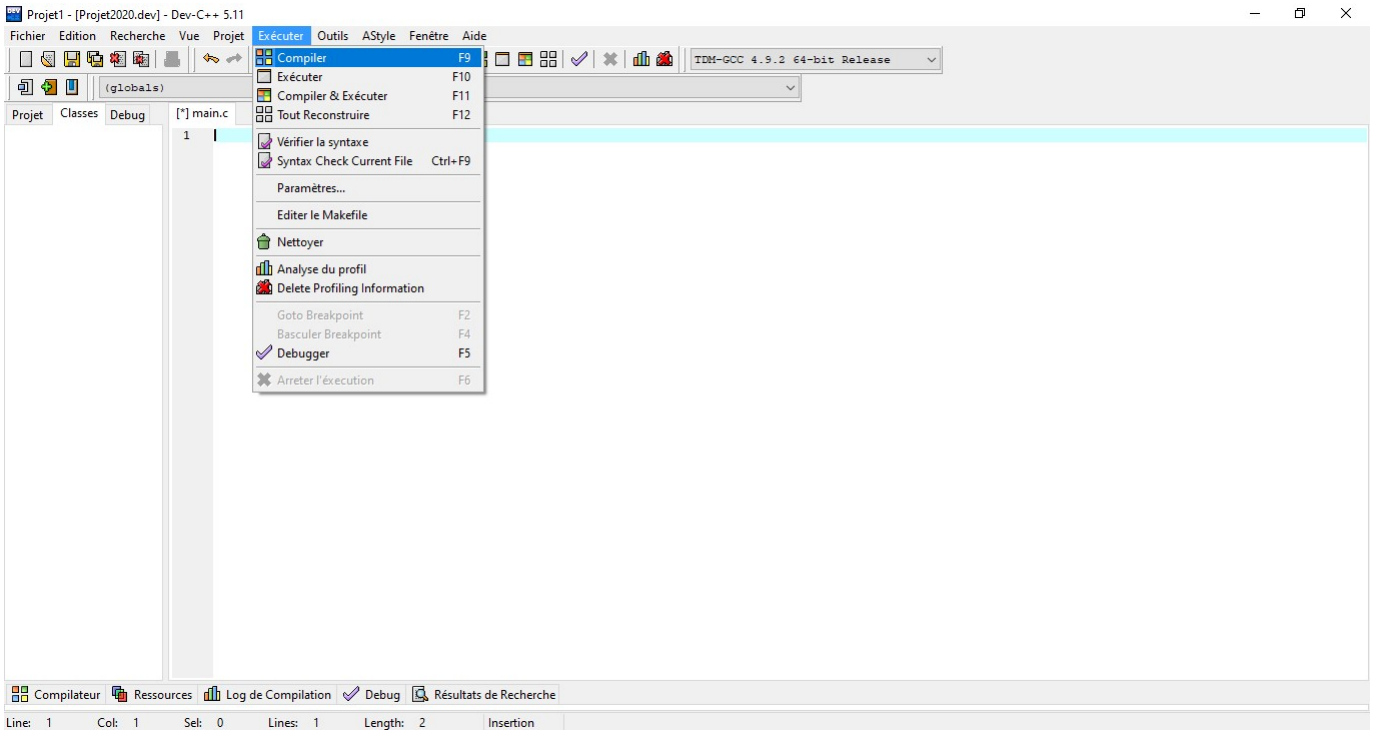
4.2.2 Fichier existant

Dans le cas où vous voulez travailler sur un fichier qui existe déjà, vous pouvez l'ouvrir dans Dev-C++ par la commande **Ouvrir Projet ou Fichier...** du menu **Fichier**. D'autre part, si Windows est bien configuré (c'est le cas, en principe, si l'installation s'est bien passée), les icônes des fichiers .CPP ressemblent à l'icône de la figure ci-dessous :

Il faut cliquer deux fois sue l'icône CPP pour lancer Dev-C++.

4.2.3 Fichier «compilé» et «exécuté»

Pour compiler le fichier, utilisez le menu **Exécuter>Compiler** ou cliquez sur le bouton avec la même icône en haut à gauche de la fenêtre suivan :



S'il y a des erreurs dans votre programme, les messages d'erreur seront affichés dans la fenêtre du compilateur (bas de la fenêtre) de la figure ci-dessous. Double-cliquez sur la première erreur : vous serez alors amenés à la ligne du programme ou se situe l'erreur.

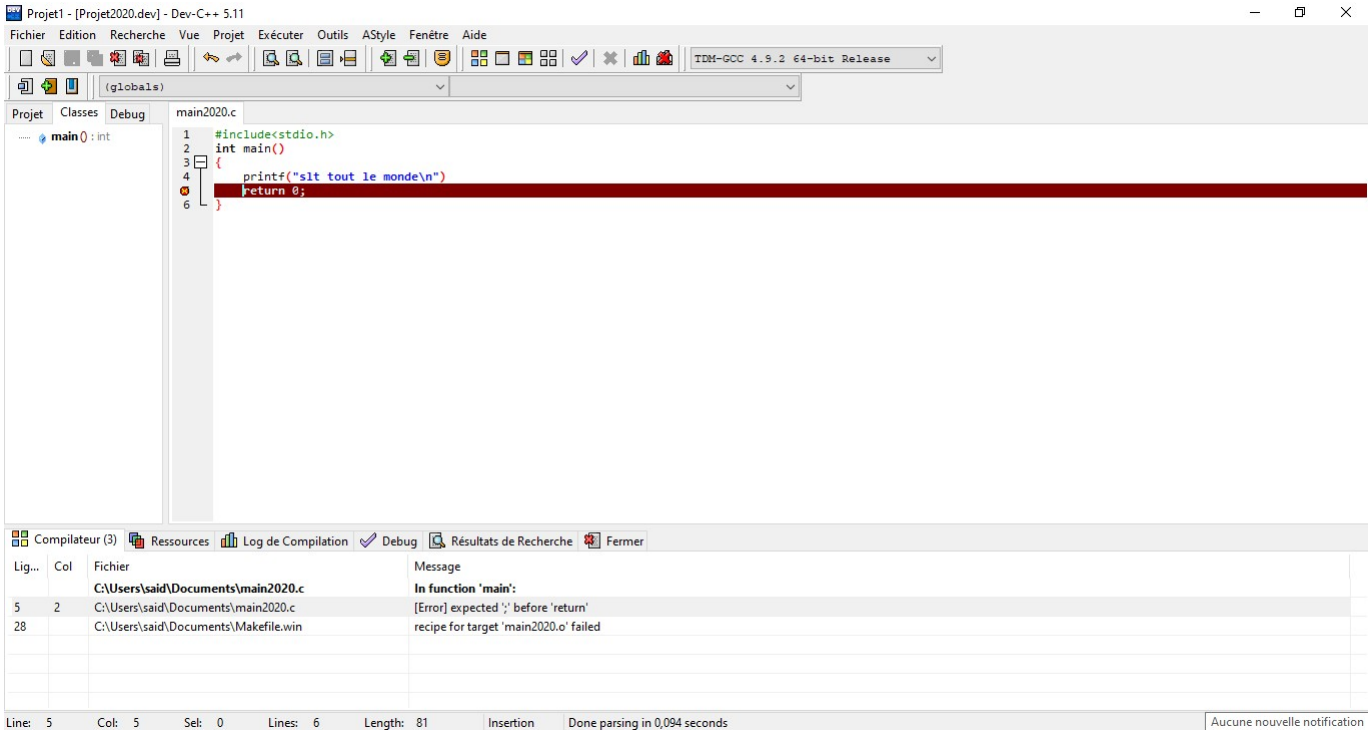
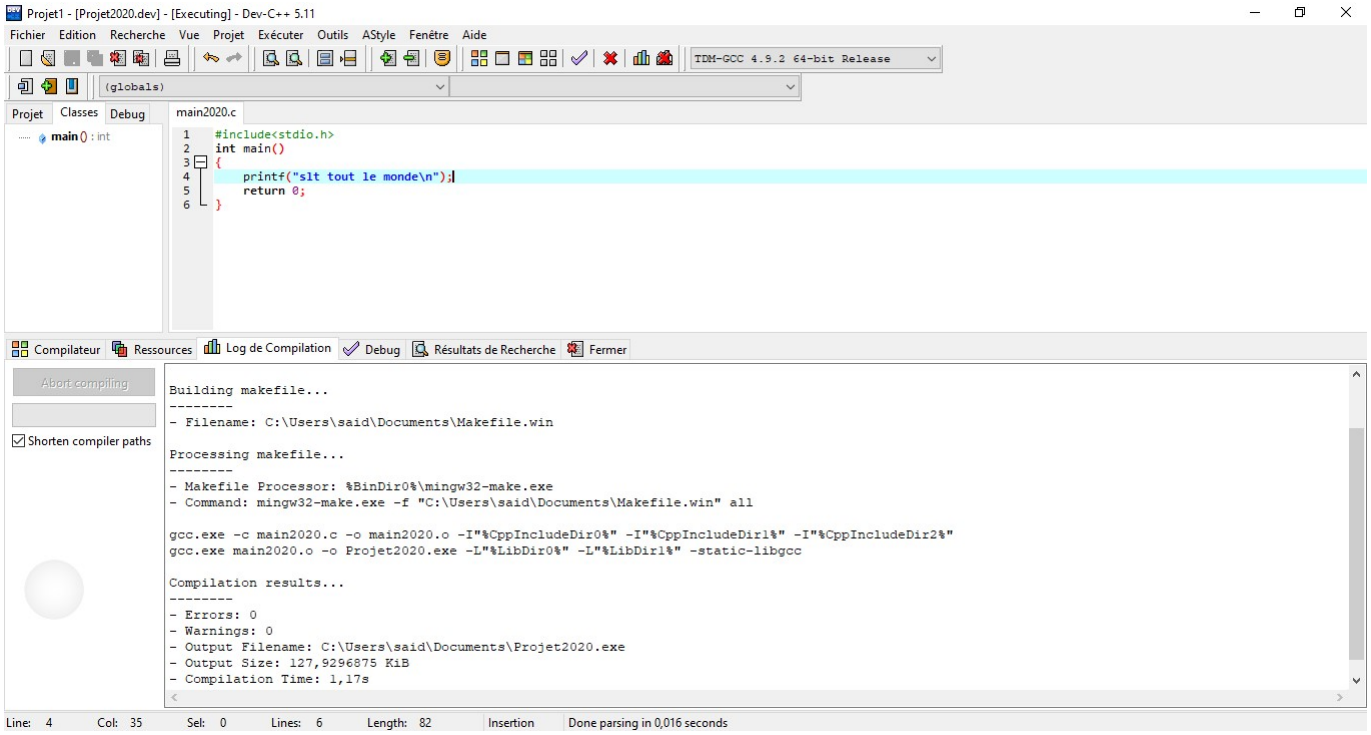


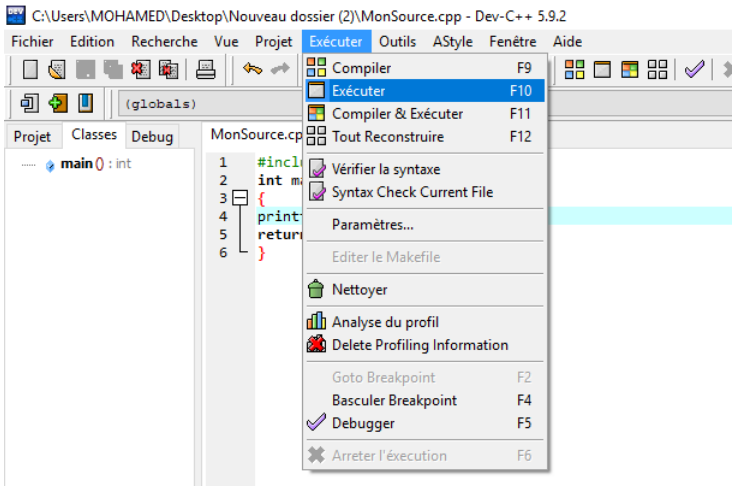
FIGURE 4.2 – Interface de Dev-C++

L'erreur signalé par une couleur spéciale et une marque dans la marge.

Une fois que les erreurs ont été corrigées la fenêtre de compilation Compile progresse nous informera que c'est terminé. Vous pouvez alors la fermer. Comme montré dans la fenêtre suivant :



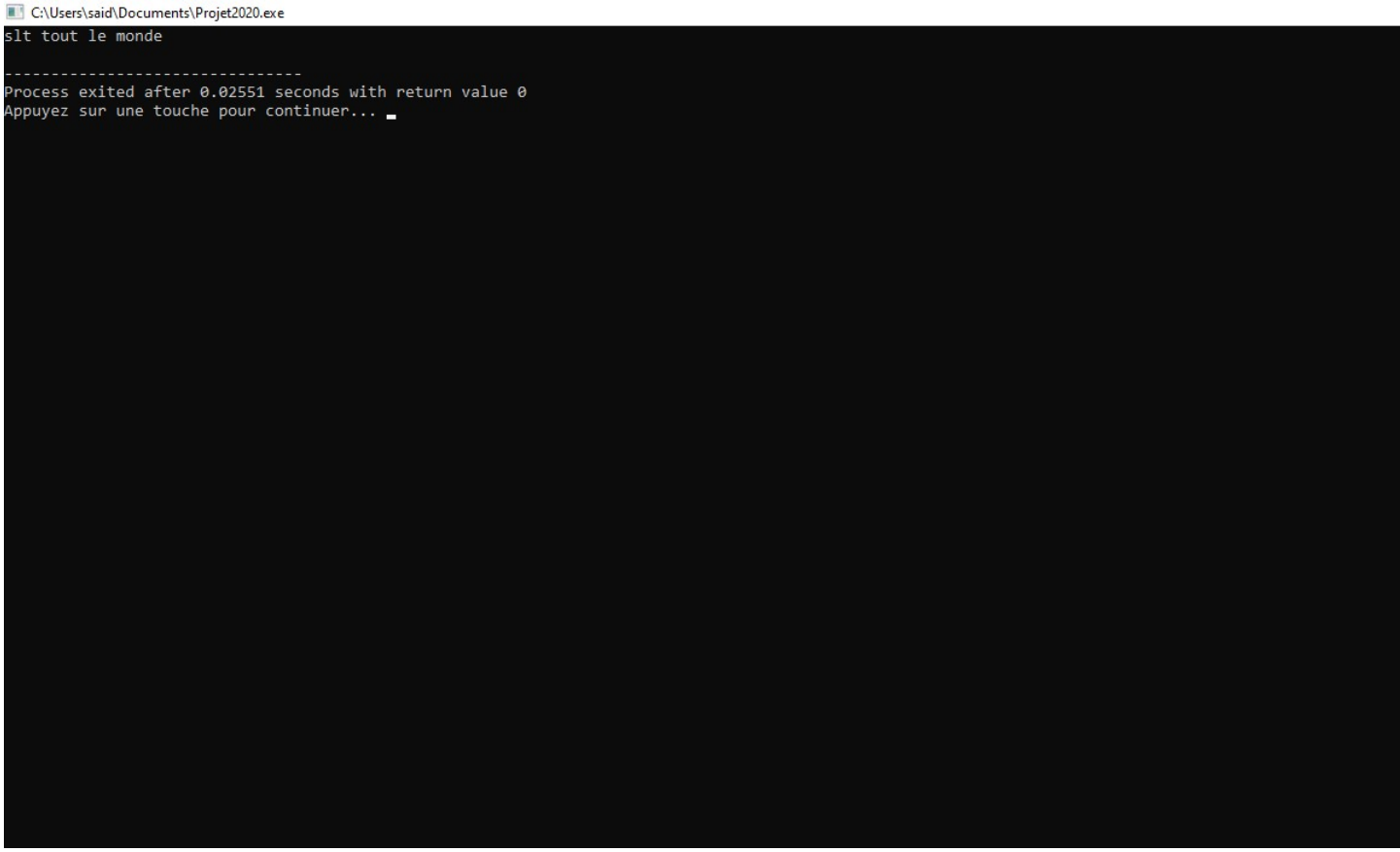
Vous pouvez alors exécuter votre programme. Pour cela utilisez le menu **Exécuter>Exécutez** ou tapez [**F10**] ou cliquez sur le bouton avec la même icône en haut à gauche.



La fenêtre console d'exécution apparaît alors :

Le programme se déroule dans la fenêtre ci-dessus. L'affichage avec **printf** et le saisie avec **scanf**. À la fin de l'exécution du programme, s'affichera «Appuyez sur une touche pour continuer....». Appuyez sur une touche, la fenêtre d'exécution se fermera alors et vous reviendrez à votre programme.

Pour compiler et exécuter directement, tapez [**F9**].



4.3 Excution de la méthode coin Nord oust et cout minimum

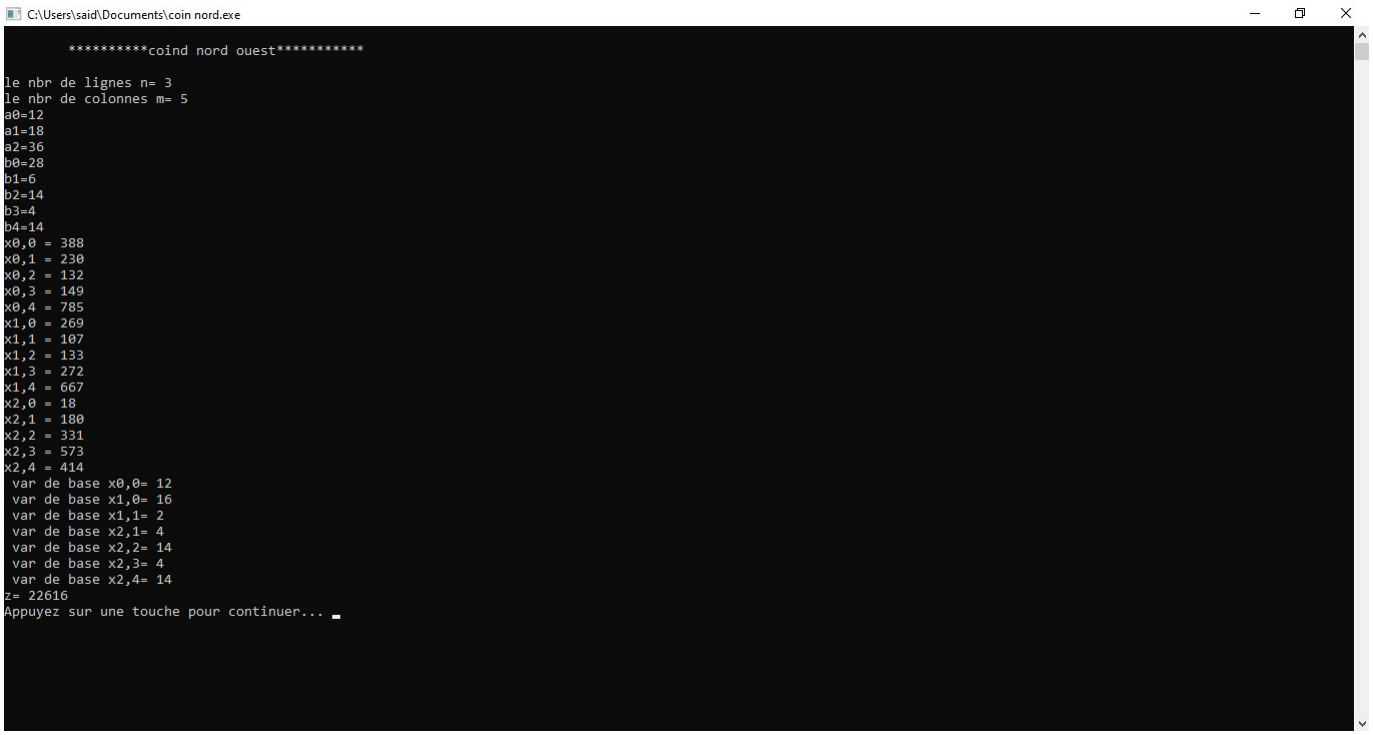
4.3.1 la Méthode coin N O

	Alger	Bejaia	jijel	Annaba	Oran	offre
Constantine	388	230	132	149	785	12
Setif	269	107	133	272	667	18
Alger	18	180	331	573	414	36
Demande	28	6	14	4	14	66

TABLE 4.1 – Problème de transport initial

• L'exécution de programme : Le programme est effectué de façon à afficher tous les détails des calculs par des étapes et ne pas pour donner le résultat seulement, on va afficher quelques résultats

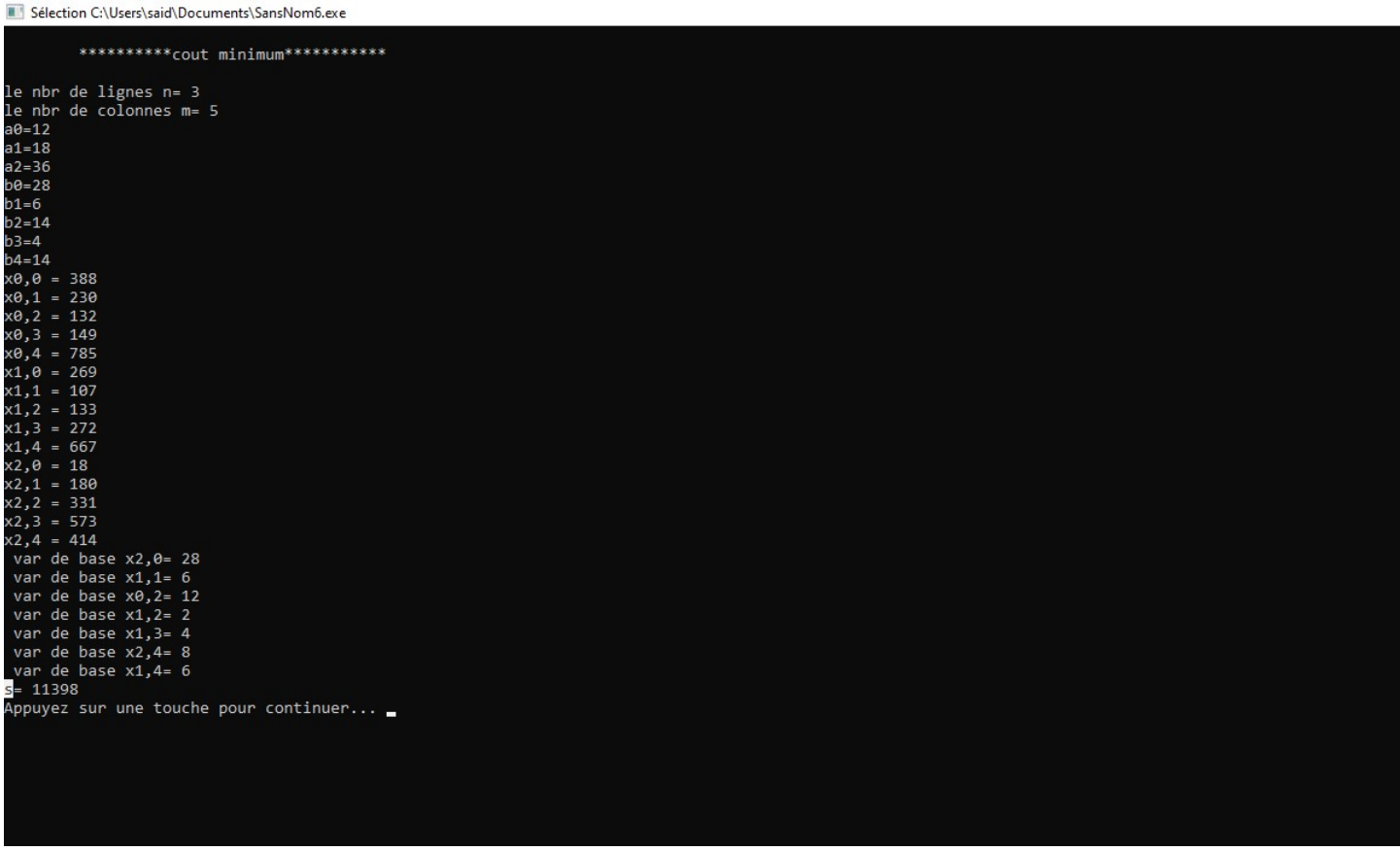
Dans les figures suivante on entrer les données, en suivant les instructions et dans la dernière instruction il nous affiche le résultat.



Certainement est 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 prés de la solution optimale.

4.3.2 Cout minimum

Dans les figures suivante on entrer les données de la Table 5.1, en suivant les instructions et dans la dernière instruction il nous affiche le résultat.



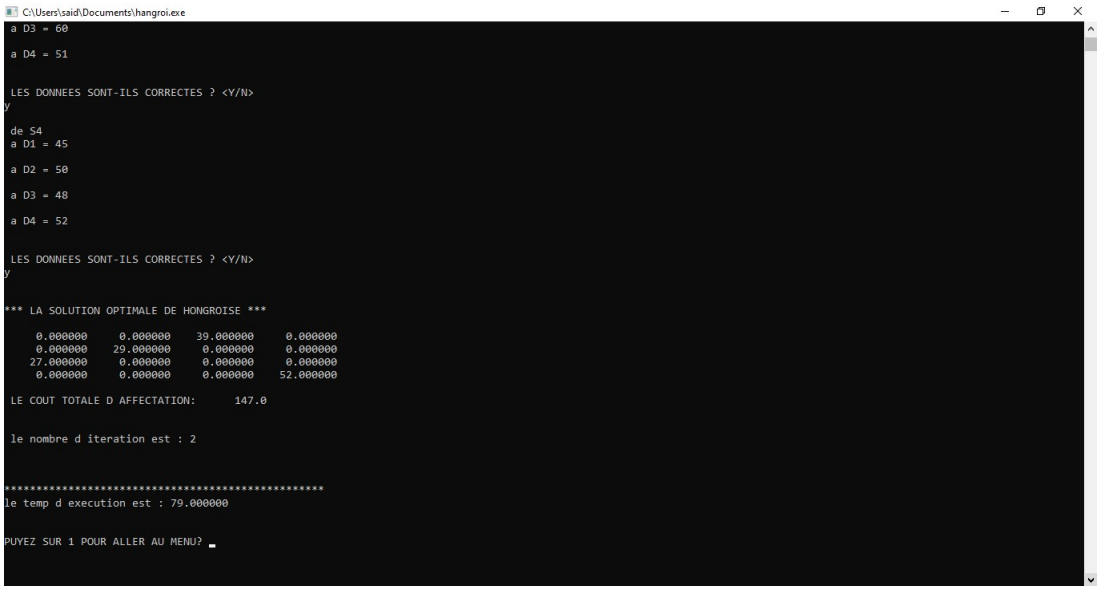
La méthode est facile à mettre en ouvre , elle trouve des solution de départ proche à la solution optimal , cette méthode est de complexité $O((m+n)^2)$, la méthode consiste à déterminer le coût le plus petits de transporter une quantité .

Complicité de la méthode Vogel

La complexité de cette méthode est $O(mnln(mn) + (m+n)^2)$. En effet, au début il faut ordonner, par valeurs croissantes, les n coûts pour chacune des m lignes et les m coûts pour chacune des n colonnes, ce qui donne $O(mnln(mn))$. Ensuite, à chaque itération et dans le pire des cas, il y a un calcul des $m+n$ différences sur les lignes et les colonnes, cela donne une complexité de $O(m+n)$ par itération. Comme il y a au plus $m+n$ itérations, le résultat suit.

4.4 Résultat d'exécution de problème d'affectation

	E	F	G	H
A	41	72	39	52
B	22	29	49	65
C	27	39	60	51
D	45	50	48	52



4.4.1 Complexité de l'algorithme

Cet algorithme est de complexité polynomiale ,complexité qu'on peut évaluer dans un premier temps en O(n4) où n est le cardinal commun des ensemble X et Y

4.5 Résultat d'exécution de problème de folt

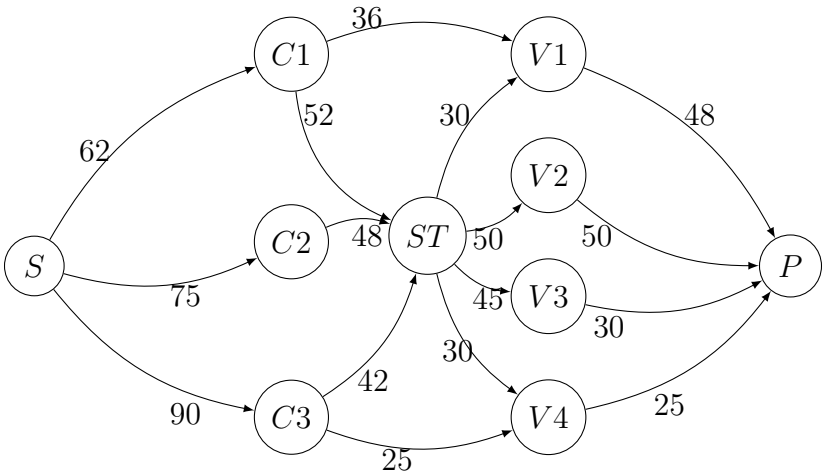


FIGURE 4.3 – problème de flot initial

```
C:\Users\said\Documents\ford et fulkerson.exe
bool: 1
iciiiii
incr: 48 5 9
incr: 36 1 5
incr: 36 0 1
9<-5<-1<-0 adds 36 incremental flow
iciiiii
incr: 49 8 9
incr: 25 3 8
incr: 25 0 3
9<-8<-3<-0 adds 25 incremental flow
iciiiii
incr: 12 5 9
incr: 12 4 5
incr: 12 1 4
incr: 12 0 1
9<-5<-4<-1<-0 adds 12 incremental flow
iciiiii
incr: 50 6 9
incr: 50 4 6
incr: 40 1 4
incr: 14 0 1
9<-6<-4<-1<-0 adds 14 incremental flow
iciiiii
incr: 36 6 9
incr: 36 4 6
incr: 36 2 4
incr: 36 0 2
9<-6<-4<-2<-0 adds 36 incremental flow
iciiiii
incr: 30 7 9
incr: 30 4 7
incr: 12 2 4
incr: 12 0 2
9<-7<-4<-2<-0 adds 12 incremental flow
iciiiii
incr: 18 7 9
incr: 18 4 7
incr: 18 3 4
incr: 18 0 3
9<-7<-4<-3<-0 adds 18 incremental flow
iciiiii
incr: 24 8 9
incr: 24 4 8
```

4.5.1 Complexité de l'algorithme

cet algorithme est de complexité théorique polynomial, complexité qu'on peut évaluer dans un premier temps en O($nm^2C\mu$). [20]

CONCLUSION

Cette étude nous a donné l'opportunité de nous 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 classiques 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 méta-heuristiques plus efficaces pour la résolution. Ce qui rend difficile de tirer une conclusion définitive sur la résolution de ce type des problèmes.

Dans notre travail nous avons essayé de résoudre trois problèmes classique de la recherche opérationnelle . Ainsi nous avons implémenté les méthodes de la recherche d'une solution de base (coin nord ouest, aproximation de VOGEL, coût minimum) et les méthodes de l'optimisation de la solution de base (Stepping-Stone, distribution modifiée) pour le problème de transport, la méthode d'hongroise pour la résolution du problème de l'affectation et les méthodes Ford-Fulkerson pour la résolution du problème de flot maximum et problème de flot maximum respectivement. Mais grâce aux développements scientifiques, différentes variations des trois problèmes ont été proposées, ainsi que différents méthodes de résolutions exactes et approchées.

Dans un premier lieu nous avons présenter quelques notions de base , puis quelques méthodes d'optimisation que nous avons appliqué à différents types de problèmes et en fin nous avons présente le logiciel Dev-C++ qui nous a permet de résoudre via une application les différentes problèmes 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] A.CHETBANI,H.BERDOUS,B.BENABBOU et S.BOUDJELDA ,La recherche des points d'articulations dans un réseau de télécommunication :Cas réseau Fibre-Optique Algérie Télécom Béjaia , mini projet, Université ABDARRAHMANE MIRA, Bejaia, 2016.

[2] BEN-IKEN MOHAMED,Problème de transport :Modélisation et résolution, MEMOIRE DE FIN D'ETUDES ,Licence Mathématiques et Applications, UNIVERSITE SIDI MOHAMED BEN ABDELLAH,2017

[3] B.BRAHMI, Cours Master 2, Recherche Opérationnelle , (2017-2018).

[4] BRUNO GARCIA, cour Recherche Opérationnelle,2000

[5] CLAUDE BERGE.Graphes et hypergraphe. North-Holland pub. Co.1973

[6] DODGE YADOLAH ,Optimisation appliquée ,Editeur : Springer Livre ,2005

[7] D.MULLER .Introduction à la théorie des graphes. (2008).

[8] F. LABURTHE. Contraintes et algorithmes en optimisation combinatoire. Thèse de doctorat, Université Paris VII- Denis Diderot, Paris, 1998.

[9] FRÉDÉRIC MEUNIER. INTRODUCTION A LA RECHERCHE OPÉRATIONNELLE. Université paris Est, CERMICS, École des ponts paristech. 2016

[10] JACQUES CARLAIRE, RECHERCHE OPÉRATIONNELLE : Optimisation Combinatoire, cours

[11] J.COHEN .Théorie des graphes et algorithmes. (oct 2006).

[12] JIN Y. WANG ,Operation Research I ,College of Management NCTU ,Fall 2008

[13] GONDRAN M.MINOUX, "Graphe et Algorithme" Eyrolles 1985

[14] L.WAYNE, WINSTON and MUNIRPALLAM VENKATARAMANAN.Introduction to Mathematical Programming : Operations Research, Volume 1 4eme édition,2003

[15] L. NTAIMO ,Transportation and Assignment Problems , INEN420 TAMU 2005

[16] M. GONDRAN et M. MINOUX Graphes et algorithmes. Eyrolles, Paris, 1995.

[17] N.BELHARRT, La Théorie des Graphes (Recherche Opérationnelle) Cour , Bouira-Algérie, 2002-2003

[18] S. SKIENA, The Algorithm Design Manual, Springer, 2ème édition, 2008.

[19] S.KEBBI,N.MAOUCHE,Affectation des vols au niveau de l'Aéroport de Bégaïa mini projet, Université ABDARRAHMANE MIRA, Bejaia, 2018.

[20] YVES DE SMET ,BERNARD FORTZ,Algorithmique 3 et Recherche Opérationnelle,2013-2014

Résume

L'objective de ce travail de montrer l'importance du problème de transport dans la résolution et l'optimisation par ces outil dont la théorie des graphe et la programmation linéaire dans les enjeux économiques , ce mémoire contribue également à montre l'importance des programme linaire et des graphe (plus particulièrement les graphe orienté sans boucle). Dans la résolution de certains problèmes de la RO, et cela en cherchant quelques problèmes d'optimisation pour les quelles nous donnons quelque algorithmes de résolution pour chaque problème suivie d'une résolution, en faisant appel à un programme réalisé sous Dev-C++ .
Mot clé : recherche opérationnelle, optimisation, programmation linéaire, théorie des graphe,problème de transport, graphe, Dev-C++

Abstract

the objective of this work to show the importance of the transport problem in the resolution and optimization by these tools including the theory of graphs and linear programming in economic issues, this thesis contributes also to show the importance of linear programs and graphs (more especially directed graph without directed loop). In the resolution of some OR problems, and this by looking for some optimization problems for which we give some algorithms of resolution for each problem followed by a resolution, using a program produced under Dev-C ++. keyword : operations research, optimization, linear programming, theory graph, transport problem, graph, Dev-C ++