

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE



جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

UNIVERSITÉ ABDERRAHMANE MIRA DE BÉJAÏA
FACULTÉ DES SCIENCES EXACTES
DÉPARTEMENT DE RECHERCHE OPÉRATIONNELLE

MEMOIRE DE FIN DE CYCLE

En vue de l'obtention du diplôme de

MASTER en Mathématique Appliquée

Option : Modélisation Mathématique et Évaluation de Performance des Réseaux

THEME

Résolution du problème de flot à coût minimum par la
méthode de support réseau

Présenté par :

M^{elle} MASSIOUN Radia
M^{elle} BENSIXHALED Nihad

Devant le jury composé de :

Président : Dr **ASLI Larbi** MCA, Université de Béjaïa.
Encadrant : Dr **BRAHMI Belkacem** MCA, Université de Béjaïa.
Examineur : D^r **SOUFIT Massinissa** MCB, Université de Béjaïa..
Examineur : D^r **TOUATI Soufiane** MCB, Université de Béjaïa.

Promotion 2022 — 2023

Remerciements

Au terme de ce travail qui marque la fin du cycle de Master de notre formation au sein de l'université Abderahmane Mira de Béjaia, nous exprimons notre gratitude à Dieu, le tout-puissant, de nous avoir accordé le savoir, le courage et la force nécessaires pour mener à bien et à terme ce travail .

*Nous remercions notre encadrant **M.BRAHMI Belkacem** pour avoir dirigé notre travail, pour tout le temps qu'il nous a consacré, pour ses précieux conseils et pour son aide.*

Nous souhaitons également remercier les membres du jury pour l'intérêt qu'ils ont porté à notre travail en acceptant de l'examiner et de l'enrichir par leurs propositions. Nous exprimons également notre gratitude envers tous nos enseignants pour leur aide et leur contribution précieuse à notre formation. Enfin, nous remercions de tout cœur tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicace

De profond de mon coeur, je dédie ce travail à tous ceux qui me sont chers.

Je souhaite exprimer ma profonde gratitude, mon amour à ma mère bien-aimée,

***Farida**, pour tout le soutien inconditionnel et l'affection qu'elle m'a accordée depuis mon enfance. J'espère que ses bénédictions m'accompagneront toujours.*

Que ce modeste travail réalise tous les vœux qu'elle a formulés et qu'il soit le fruit de ses innombrables sacrifices. Que Dieu, le Très-Haut, lui accorde santé, bonheur et une longue vie épanouissante.

*À la mémoire de mon très cher papa **Rachid**, décédé trop tôt, qui a été toujours à mes côtés pour me soutenir et m'encourager.*

*À Mes adorables petits frères **Abd louahab** et **Manel**, je prie pour qu'ils soient protégés et qu'ils connaissent la chance et le bonheur dans leur vie.*

*À mes chers enseignants , M. **BRAHMI Belkacem**, M. **ASLI Larbi**, merci pour votre soutien.*

*À mes chères copines **Tinhinane**, **Hadda**, **Asma**, **Djidji**, **Sarah**, **chahinez**, **Romaissa**, **Khalida**, avec qui j'ai partagé les meilleurs moments de complicité et de joie.*

*À ma binôme **Radia**, à qui je souhaite la réussite.*

À tous ceux qui me sont chers, à vous tous.

Merci.

- Nihad

Dédicace

De profond de mon coeur, je dédie ce travail à tous ceux qui me sont chers.

*Je dédie ce travail à la mémoire de mes chers parents bien-aimés, **Nouria** et **Lakhdar**, qui ont été une source d'amour, de soutien et d'inspiration tout au long de ma vie. Leur présence et leurs encouragements me manquent chaque jour, mais je sais qu'ils veillent sur moi depuis les cieux. Ce travail est un témoignage de leur dévouement et de leur impact positif sur ma vie.*

À mes frères bien-aimés Souhila, Fouzia, Celia, Meriam, Mourad, Zoulikha et Samira, vous êtes ma source de joie et de bonheur.

*À mes chers enseignants , M. **BRAHMI Belkacem**, M. **ASLI Larbi**, M. **CHABANE**, M. **KABYLE Kamel**, merci pour votre soutien.*

À ma merveilleuse copine Nihad, merci pour votre amour et votre soutien précieux.

*A mes camarades de la promotion **RO** votre soutien et vos moments partagés resteront gravés dans ma mémoire.*

Enfin, à mes chers amis, Saadiya, Thilleli, Sohieb, Mosaab, Mounir, Riad, merci d'avoir été là à chaque étape de ma vie.

Merci.

- Radia

TABLE DES MATIÈRES

| | |
|--|-----------|
| 1 INTRODUCTION À L'OPTIMISATION DANS LES RÉSEAUX | 8 |
| 1.1 Généralités sur les graphes | 9 |
| 1.1.1 Graphe non orienté | 9 |
| 1.1.2 Graphe orienté | 10 |
| 1.1.3 Sous-graphe | 13 |
| 1.1.4 Représentations matricielle des graphes | 14 |
| 1.2 Optimisation dans les réseaux | 16 |
| 1.3 Quelques problèmes d'optimisation dans les réseaux | 16 |
| 1.3.1 Problème de plus court chemin | 16 |
| 1.3.2 Le problème du flot maximum | 18 |
| 1.3.3 Problème d'arbre couvrant | 18 |
| 2 PROBLÈME DE FLOT À COÛT MINIMUM | 20 |
| 2.1 Problèmes de flots dans les réseaux | 20 |
| 2.1.1 Problème de flot général | 21 |
| 2.1.2 Problème de flot à coût minimal | 21 |
| 2.2 Méthode de simplexe réseau | 22 |
| 2.2.1 Définition et concepts de base | 23 |
| 2.3 Exemple pratique | 25 |
| 3 Méthode directe de support | 32 |
| 3.1 Position du problème et définitions | 33 |
| 3.1.1 Formule d'accroissement de la fonction objectif | 33 |

| | |
|---|-----------|
| 3.1.2 Critère d'optimalité | 34 |
| 3.1.3 Estimation de suboptimalité | 35 |
| 3.1.4 Algorithme de résolution | 35 |
| 3.2 Algorithme de support | 38 |
| 4 Application de la méthode de support sur les réseaux | 41 |
| 4.1 Méthode de support réseau | 42 |
| 4.1.1 Vecteur des coûts réduits | 42 |
| 4.1.2 Critère d'optimalité | 43 |
| 4.2 Exemple pratique | 45 |
| 4.3 Exemple d'application | 51 |
| 4.3.1 Construction du modèle | 52 |
| 4.3.2 Interprétation des résultats | 54 |

TABLE DES FIGURES

| | |
|--|----|
| 1.1 Graphe non orienté | 9 |
| 1.2 Un digraphe et Un graphe sous-jacent de G | 11 |
| 1.3 Graphe connexe | 12 |
| 1.4 Un graphe biparti complet $K_{2,3}$ et Un graphe complet K_4 | 12 |
| 1.5 Arbre et Une arborescence de racine c. | 13 |
| 1.6 Illustrations d'un Graphe G,sous-graphe induit,sous-graphe partiel et Le complé- mentaire | 14 |
| 1.7 Graphe orienté et sa matrice d'incidence | 14 |
| 1.8 Graphe orienté et sa matrice d'incidence | 15 |
| 1.9 Graphe non orienté et sa matrice d'incidence | 16 |
| 1.10 Exemple d'un graphe | 18 |
| 2.1 Réseau de transport | 25 |
| 2.2 Choix de l'arbre couvrant | 26 |
| 2.3 Choix de la variable entrante | 27 |
| 2.4 Le nouveau marquage | 28 |
| 2.5 Choix de la variable entrante (itération 2) | 28 |
| 2.6 Mise à jour des potentiels | 29 |
| 2.7 Choix de la variable entrante (itération 3) | 30 |
| 2.8 La solution optimal | 31 |
| 4.1 Réseau de distribution | 45 |
| 4.2 Choix de l'arbre couvrant | 46 |
| 4.3 Variable entrante | 47 |

| | | |
|-----|--|----|
| 4.4 | Le nouveau marquage | 47 |
| 4.5 | Variable entrante | 48 |
| 4.6 | Mise à jour des potentiels | 49 |
| 4.7 | La variable entrante | 50 |
| 4.8 | La solution optimal | 51 |
| 4.9 | Réseau de distribution de l'entreprise | 52 |

INTRODUCTION GÉNÉRALE

La recherche opérationnelle (RO) est une discipline des mathématiques appliquées qui vise à résoudre des problèmes complexes de prise de décision dans divers domaines, tels que : la logistique, la gestion de la chaîne d'approvisionnement, la planification de la production, les transports, \dots etc. L'un des problèmes classiques étudiés en recherche opérationnelle est le problème de flot à coût minimum qui concerne l'optimisation du flux des ressources dans un réseau.

Le problème de flot à coût minimum (PFCM) consiste à déterminer les quantités (flux) de la ressource à acheminer de la source vers la destination de telle sorte à minimiser le coût total de transport dans le réseau. En pratique, pour un réseau donné, la ressource peut être une marchandise (un ou des produit(s)) pour le cas d'un réseau de distribution, de l'énergie pour le cas d'un réseau électrique ou des informations pour le cas d'un réseau informatique ou de télécommunication. La ressource doit être acheminée depuis une ou des sources vers une ou des destinations via des arcs inter-connectant les nœuds du réseau. Chaque arc a une capacité maximale et un coût unitaire associé au transport de la ressource tout au long de cet arc [20].

Il existe une grande variété de méthodes de résolution du PFCM, dont les plus populaires sont les méthodes primales du simplexe [9] et les méthodes primales-duales [3, 20]. L'algorithme du simplexe réseau est l'approche couramment employée pour résoudre le PFCM. Ce dernier est une extension de l'algorithme du simplexe classique de programmation linéaire. L'algorithme du simplexe réseau est spécifiquement adapté aux problèmes d'optimisation dans les réseaux, car la matrice des contraintes est creuse. Les propriétés particulières de cette matrice ont permis d'implémenter efficacement cette approche et de résoudre des problèmes pratiques de grande taille.

L'objectif principal de notre mémoire est l'application de la méthode du support dans les réseaux, et plus précisément résoudre le problème de flot à coût minimum. Dans ce travail, nous

allons adapté la méthode directe de support de programmation linéaire, développé par R. Gabasov et al. [13] pour résoudre le PFCM. L'avantage de cette approche est de traiter les problèmes tels qu'ils sont formulés initialement sans les transformer. Un autre avantage est d'utiliser le critère de sub-optimalité pour arrêter le processus de résolution à une précision donnée par l'utilisateur.

Notre mémoire est composé d'une introduction générale, quatre chapitres, une conclusion générales et une bibliographie.

Dans le premier chapitre, nous rappelons les notions de base de la théorie des graphes et une introduction à l'optimisation dans les réseaux. Le deuxième chapitre traite le problème de flot à coût minimum, ainsi qu'une présentation de la méthode du simplexe réseau. Dans le troisième chapitre, nous présentons la méthode directe de support pour la résolution des problèmes classiques de programmation linéaire. Dans le dernier chapitre, nous avons étendu la méthode directe de support pour la résolution du problèmes de flot à coût minimum. Nous terminons ce travail par une étude de cas pour illustrer l'application de la méthode du support réseau (MSR). Pour ce faire, nous avons implémenté sur Matlab l'algorithme MSR.

CHAPITRE 1

INTRODUCTION À L'OPTIMISATION DANS LES RÉSEAUX

Introduction

L'optimisation des réseaux avec la théorie des graphes vise à trouver la meilleure configuration des flux et des ressources à travers le réseau pour atteindre un objectif spécifique. Cet objectif peut être de minimiser les coûts de transport, de maximiser la capacité du réseau, de minimiser les temps de trajet ou de répondre aux demandes des utilisateurs de manière optimale.

Dans ce chapitre, nous présentons les concepts fondamentaux de la théorie des graphes et une introduction à l'optimisation dans les réseaux. Par la suite, nous explorons les problèmes classiques d'optimisation dans les réseaux, à savoir les problèmes de : cheminement, flot maximal, arbre couvrant de poids maximum et le problème de flot à coût minimum qui l'objet de notre travail. Nous donnons la formulation mathématique de ces problèmes et quelques algorithmes de résolution.

1.1 Généralités sur les graphes

Un **graphe** est une structure mathématique qui représente un ensemble d'objets et les relations entre eux. Un graphe est composé de sommets (ou nœuds) et d'arêtes (ou liens), où chaque arête relie deux sommets [7].

1.1.1 Graphe non orienté

Un **graphe non orienté** fini G est défini par un couple de deux ensembles (V_G, E_G) , V_G est un ensemble non vide de **sommets** (vertices) de G et E_G désigne l'ensemble de ses **arêtes** (edges), on écrit $G = (V_G, E_G)$ ou simplement $G = (V, E)$ s'il n'y a pas d'ambiguïté. un sommet v est représenté par un point $p(v)$ dans le plan ou dans l'espace, tandis que une arête e est définie par un couple de deux sommets u et v , elle est représentée par une courbe reliant les deux points $p(u)$ et $p(v)$, on la note $e = (u, v)$ ou $e = uv$. Notons que uv et vu représentent la même arête. Les sommets u, v sont les extrémités (endpoints) de l'arête $e = uv$, si $u = v$, l'arête $e = uu$ est une **boucle** (loop), si deux sommets sont reliés par plusieurs arêtes, G est un **multi graphe** (multigraph) [2].

Définition 1.1. Graphe simple

Un graphe G est dit **simple** (simple graph) lorsque il ne contient ni boucle ni arêtes multiples. Le nombre de sommets et d'arêtes de G sont respectivement notés n et m . Ces deux nombres sont appelés respectivement **l'ordre** et la **taille** de G . Tous les graphes utilisés dans ce mémoire sont simples et finis (les paramètres n et m sont finis). La figure ci- dessous présente le diagramme d'un graphe d'ordre 6, simple et non orienté. $V_G = \{a, b, c, d, e, f\}$ et $E_G = \{ac, ad, ae, bc, be, bf, ce, cf, df\}$.

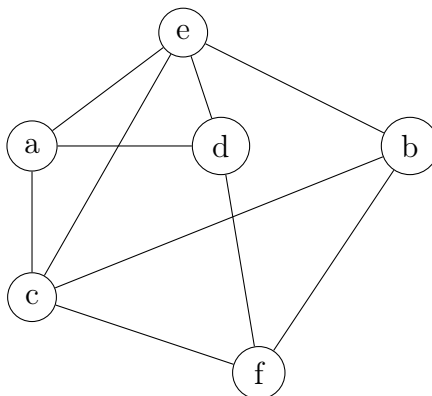


FIGURE 1.1 – Graphe non orienté

Les extrémités d'une arête sont dites **incidentes** à cette arête, et vice versa. Deux sommets

incidentes à une même arête sont **adjacents** (adjacent vertices) ou **voisins** (neighbors); de même, deux arêtes incidentes au même sommet sont dites adjacentes. L'ensemble des sommets adjacents à v , excepte lui-même, est appelé **voisinage ouvert** (open neighborhood) de v , il est noté $N(v) = \{u \in V, vu \in E\}$, le voisinage **fermé** (closed neighborhood) d'un sommet v , noté $N[v] = N(v) \cup \{v\}$ [16].

Définition 1.2. Degré d'un graphe non orienté

Le **degré** (degree) d'un sommet v , noté $d_G(v)$, est égal au nombre d'éléments dans $N(v)$, c'est-à-dire $d_G(v) = |N(v)|$. Un sommet de degré nul est dit **isolé** (isolated vertex). Le plus petit degré d'un graphe G est noté $\delta(G) = \min_{v \in V} d(v)$ et son plus grand degré est noté $\Delta(G) = \max_{v \in V} d(v)$ ils sont, respectivement, égaux au degré d'un sommet ayant le moins et le plus de voisins.

Dans la figure précédente, le sommet a est adjacent aux sommets c, d et e donc $N(a) = \{c, d, e\}$ et $N[a] = \{a, c, d, e\}$. les arêtes ac et cf sont adjacentes. $\Delta(G) = d(c) = 4$, $\delta(G) = d(a) = 3$.

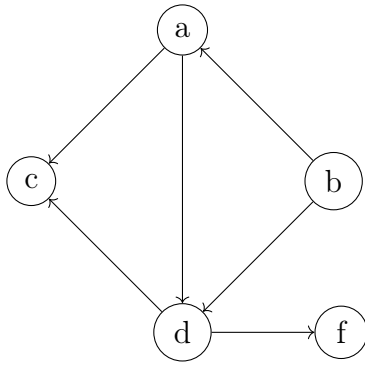
Définition 1.3. Graphe régulier

Un graphe est **régulier** si tous ses sommets ont le même degré, c'est-à-dire, $\delta(G) = \Delta(G) = k$, on dit alors que le graphe est régulier de degré k ou k -régulier.

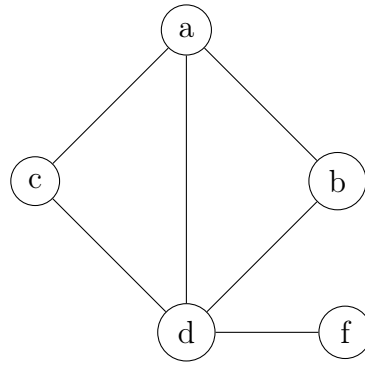
1.1.2 Graphe orienté

Un **graphe orienté** G ou **digraphe** (directed graph ou digraph) est un couple (V_G, U_G) formé d'un ensemble de sommets V_G et d'un ensemble de couples ordonnés dont les éléments sont appelés les arcs de V_G . Ainsi, à tout arc a correspond un couple ordonné de deux sommets (u, v) tel que $a = (u, v)$, u est l'extrémité initiale de l'arc a , noté $I(a) = u$ et v son extrémité terminale, noté $T(a) = v$. Graphiquement, l'arc (u, v) est représentée par une courbe reliant les deux sommets u et v , orientée de u vers v (voir la figure 1.2). A tout digraphe G nous pouvons associer un graphe g avec le même ensemble de sommets en remplaçant chaque arc par une arête avec les mêmes extrémités. Ce graphe est le graphe **sous-jacent** de G noté, $G(g)$ A l'inverse, tout graphe G peut être vu comme un digraphe, en remplaçant chaque arête par deux arcs d'orientations opposées avec les mêmes extrémités, ce digraphe est le digraphe associé à G , noté $G(g)$.

Tout concept valide pour les graphes non orientés, excepte ceux pour lesquels l'orientation joue un rôle essentiel, s'applique automatiquement aux digraphes. Par exemple, le degré d'un sommet u dans un digraphe G est simplement le degré de u dans le graphe sous-jacent de G , un digraphe G est simple si son graphe sous-jacent g est simple.



Un digraphe



Un graphe sous-jacent de G

FIGURE 1.2 – Un digraphe et Un graphe sous-jacent de G

Pour tout sommet x de G , on associe :

- $d_G^+(x) = \{a \in U, I(a) = x\}$ le demi-degré extérieur de x , c'est le nombre d'arcs ayant x comme extrémité initiale.
- $d_G^-(x) = \{a \in U, T(a) = x\}$ le demi-degré intérieur de x , c'est le nombre d'arcs ayant x comme extrémité terminale.
- $d_G(x) = d_G^-(x) + d_G^+(x)$ le degré de x , c'est le nombre d'arcs ayant x comme extrémité.

Remarque 1.1. Comme pour les graphes orientés, on dit que l'arc $a = (x, y)$ est incident à x et y et que x et y sont adjacents. Une boucle est un arc a tel que $I(a) = T(a)$.

Définition 1.4. Chemin et chaîne

Un **Chemin** dans un graphe est une séquence de sommets dans laquelle chaque sommet est relié au suivant par une arête. Soit $G = (V, E)$ un graphe non orienté, où V est l'ensemble des sommets et E est l'ensemble des arêtes. Un chemin de longueur k de u à v dans G est une séquence de sommets $v_0, v_1, v_2, \dots, v_k$ telle que $v_0 = u$, $v_k = v$ et $(v_i, v_{i+1}) \in E$ pour tout i de 0 à $k - 1$.

Une **Chaîne** est un chemin dans lequel tous les sommets sont distincts. Une chaîne simple est une chaîne dans laquelle tous les sommets sont distincts, à l'exception du premier et du dernier sommet. En d'autres termes, aucune répétition de sommets n'est autorisée sauf pour le premier et le dernier sommet de la chaîne.

Graphe connexe Graphe dans lequel on peut relier, directement ou non, n'importe quel sommet à n'importe quel autre sommet du graphe par une chaîne d'arêtes, i.e que à partir de chacun des sommets de ce graphe, on peut se rendre à n'importe quel autre sommet de ce graphe [6].

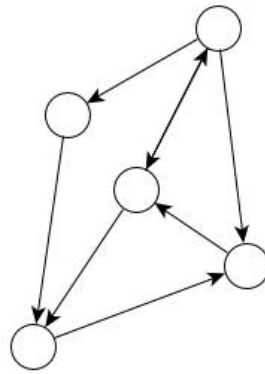


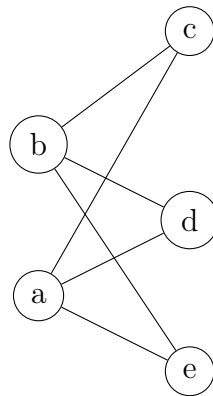
FIGURE 1.3 – Graphe connexe

Définition 1.5. Graphe complet

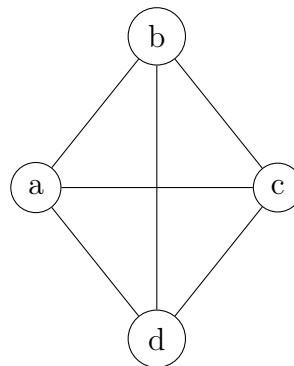
Un graphe G est **complet** (complet graph) si entre deux sommets quelconques, il y a une arête qui les relie. On note par K_n le graphe complet d'ordre n .

Définition 1.6. Graphe pondéré

Un graphe **pondéré**, un graphe (orienté ou non) dont les arêtes ont été affectées d'un nombre appelé poids (ou coût).



Un graphe biparti complet $K_{2,3}$



Un graphe complet K_4

FIGURE 1.4 – Un graphe biparti complet $K_{2,3}$ et Un graphe complet K_4

Définition 1.7. Arbre

Un **arbre** (tree) est un graphe connexe sans cycle. La proposition suivante caractérise un arbre.

Proposition 1.1. Soit $G = (V, E)$ un graphe, avec $n = |V|$ et $m = |E|$, les propriétés suivantes sont équivalentes :

1. G est un arbre ;

2. G est sans cycle et possède $n - 1$ arêtes ;
3. G est connexe et possède $n - 1$ arêtes ;
4. G est sans cycle et si on lui ajoutant une arêtes, on crée un et un seul cycle ;
5. G est connexe et si on lui supprime une arête, il n'est plus connexe ;
6. Il existe une chaîne et une seule entre toutes paires de sommets.

Une **arborescence** est un graphe orienté sans circuit admettant une racine (ou une source) $s \in V$, c'est à-dire, un sommet s vérifiant la propriété suivante : pour tout autre sommet $x \in V$, il existe un chemin unique allant de s à x [23].

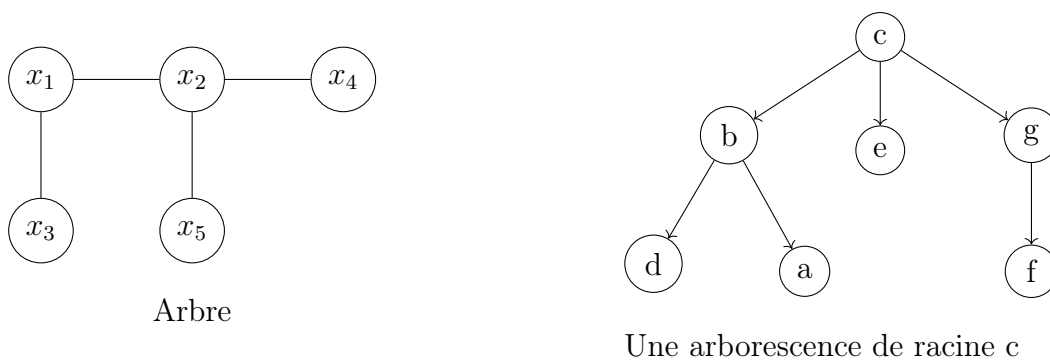
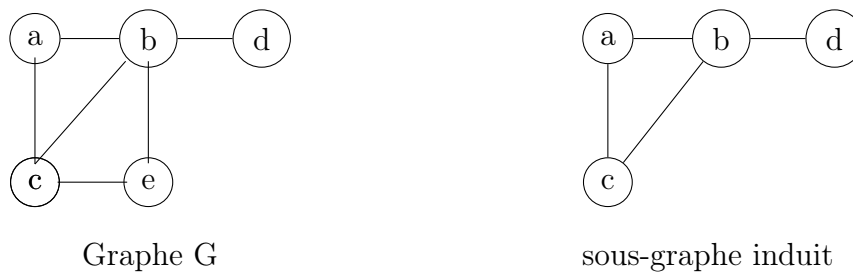
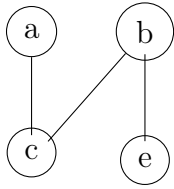


FIGURE 1.5 – Arbre et Une arborescence de racine c

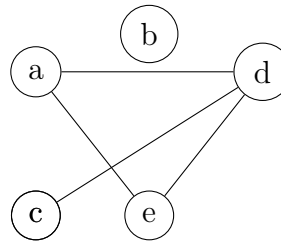
1.1.3 Sous-graphe

Pour un graphe $G = (V, E)$ on a un sous-ensemble W de sommets, notons $E(W)$ l'ensemble des arêtes de G ayant leur deux extrémités dans W . Le graphe $G' = (W, E(W))$, aussi noté $G[W]$ est appelé **sous graphe** de G **induit** par W . Soit F un sous-ensemble quelconque d'arêtes dans $E(W)$. Le graphe $G'' = (W, F)$ est appelé **sous-graphe partiel** de G . Tout sous-graphe induit est donc partiel, mais le contraire n'est pas nécessairement vrai. Le graphe **complémentaire** de G est le graphe ayant exactement les mêmes sommets que G et dans lequel deux sommets sont reliés par une arête si et seulement si ils ne le sont pas dans G [25].





sous-graphe partiel



Le complémentaire

FIGURE 1.6 – Illustrations d’un Graphe G, sous-graphe induit, sous-graphe partiel et Le complémentaire

1.1.4 Représentations matricielle des graphes

Matrice d’adjacences (matrice d’incidence sommets - sommets)

On peut représenter un graphe par une matrice d’adjacences. Une matrice $n \times n$ est un tableau de n lignes et n colonnes. (i, j) désigne l’intersection de la ligne i et de la colonne j . Dans une matrice d’adjacences, les lignes et les colonnes représentent les sommets du graphe. Un « 1 » à la position (i, j) signifie qu’un arc part de i pour rejoindre j .

Voici la matrice d’adjacences de graphe suivant :

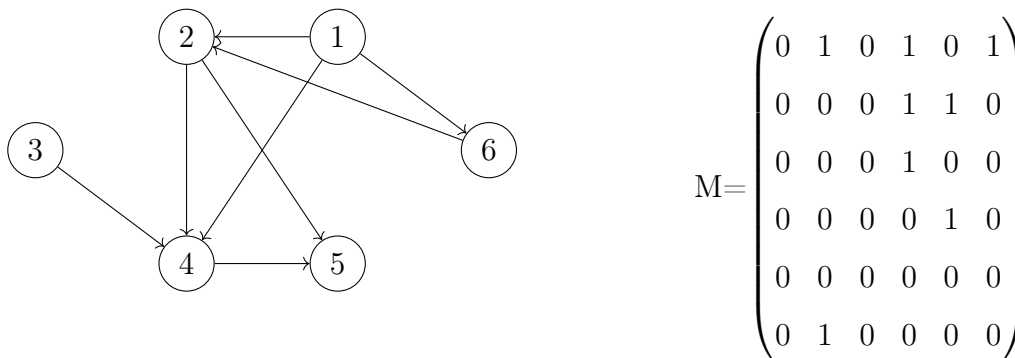


FIGURE 1.7 – Graphe orienté et sa matrice d’incidence

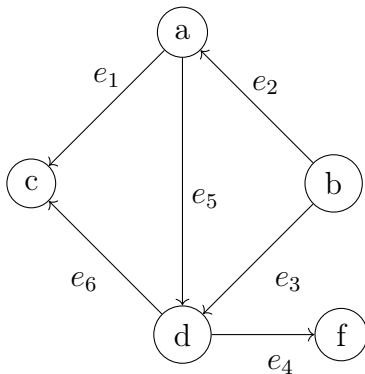
Cette matrice a plusieurs caractéristiques :

1. Elle est carrée : il y a autant de lignes que de colonnes.
2. Contrairement à celle d’un graphe non orienté, elle n’est pas symétrique.
3. Une fois que l’on fixe l’ordre des sommets, il existe une matrice d’adjacences unique pour chaque digraphe. Celle-ci n’est la matrice d’adjacences d’aucun autre digraphe [7].

Matrice d'incidence sommet arc (Matrice d'incidence aux arcs)

La matrice d'incidence sommets-arcs est une matrice $n \times m$ associée à un graphe orienté et qui est définie par [19] :

$$b_{ij} = \begin{cases} +1, & \text{si } x_i \text{ est l'extrémité initiale de } e_j \\ -1, & \text{si } x_i \text{ est l'extrémité terminal de } e_j \\ 0 & \text{sinon.} \end{cases}$$



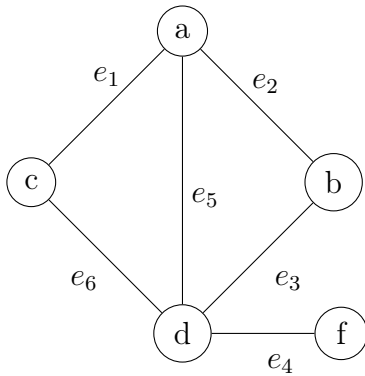
| | | | | | | |
|-----|-------|-------|-------|-------|-------|-------|
| | e_1 | e_2 | e_3 | e_4 | e_5 | e_6 |
| a | 1 | -1 | 0 | 0 | 1 | 0 |
| b | 0 | 1 | 1 | 0 | 0 | 0 |
| c | -1 | 0 | 0 | 0 | 0 | 1 |
| d | 0 | 0 | -1 | 1 | -1 | 1 |
| f | 0 | 0 | 0 | -1 | 0 | 0 |

FIGURE 1.8 – Graphe orienté et sa matrice d'incidence

Matrice d'incidence sommets-arêtes d'un graphe non orienté

La matrice d'incidence sommet-arêtes est une matrice $n \times m$ associée à un graphe non orienté et qui est définie par [17] :

$$a_{ij} = \begin{cases} 1 & \text{si le sommet } i \text{ est l'une des extrémité de l'arêtes } u_j \\ 0 & \text{sinon.} \end{cases}$$



$$b_{ij} = \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline a & 1 & 1 & 0 & 0 & 0 & 0 \\ b & 0 & 1 & 1 & 0 & 0 & 0 \\ c & 1 & 0 & 0 & 0 & 0 & 1 \\ d & 0 & 0 & 1 & 1 & 1 & 1 \\ f & 0 & 0 & 0 & 1 & 0 & 0 \end{array}$$

FIGURE 1.9 – Graphe non orienté et sa matrice d’incidence

1.2 Optimisation dans les réseaux

L’optimisation dans les réseaux est un domaine de recherche qui vise à trouver des solutions optimales pour des problèmes liés aux réseaux, en prenant en compte les contraintes associées à ces réseaux. Les méthodes d’optimisation couramment utilisées dans les réseaux incluent la programmation linéaire, la programmation entière mixte, la programmation en nombres entiers, la théorie des files d’attente, la théorie des jeux, etc [1].

Définition 1.8. Réseau

Un réseau est un graphe valué, connexe, sans boucle et ayant plus d’un sommet, noté $G = (V, E, u)$ dans lequel il y a un unique sommet S appelé la source tel que $d^-(s) = 0$ un unique sommet P appelé puit tel que $d^+(s) = 0$, pour $a \in E$ le nombre $u(a)$ est appelé capacité de l’arc [24].

On appelle noeud d’un réseau un sommet qui a plus de deux arcs incidents. On appelle branche tout chemin pour lequel seuls les premiers et derniers sommets sont des noeuds.

1.3 Quelques problèmes d’optimisation dans les réseaux

Il existe plusieurs problèmes d’optimisation. Considérons les problèmes suivants :

1.3.1 Problème de plus court chemin

Le problème de recherche d’un chemin de longueur minimale entre deux noeuds s et t d’un réseau à condition que il n’y a pas de circuit de longueur négative et il existe alors de nombreux

algorithmes efficaces pour le résoudre.

Soit un réseau $G = (V, E)$, où $V = \{1, 2, \dots, n\}$ est l'ensemble des nœuds et E celui des arcs. On associe à chaque arc $(i, j) \in E$ une longueur c_{ij} .

Le problème de plus court chemin (PCC) entre les nœuds s et t est modélisé par le programme linéaire (PL) ci-après, où les variables de décision sont définies comme suit :

$$x_{ij} = \begin{cases} 1, & \text{si on emprunte l'arc } (i, j), \\ 0 & \text{sinon} \end{cases}$$

$$\min \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (1.1)$$

sujet à :

$$\sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1, & i = s, \\ 0, & i \neq s, i \neq t, \\ -1, & i = t, \end{cases} \quad (1.2)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in E \quad (1.3)$$

Un chemin de longueur μ minimale entre les nœuds s et t peut être obtenu à partir de la solution optimale x^* du problème (1.2)-(1.3) et ce en posant $\mu = x_{st}^* = \{(i, j) \in E \mid x_{ij}^* = 1\}$.

Il existe plusieurs algorithmes efficaces pour résoudre le problème du PCC, on cite :

- **Algorithme de Dijkstra**

Il permet de calculer le plus court chemin entre un sommet particulier et tous les autres dans un graphe pondéré dont tous les poids sont positifs. L'algorithme comporte une phase d'initialisation. À chaque sommet on attribue un poids qui vaut 0 pour le sommet de départ et infini pour les autres sommets.

Le traitement de l'algorithme consiste à examiner les sommets les uns après les autres et à sélectionner le sommet x auquel on a affecté la plus petite distance du sommet de départ jusqu'à x . On recommence tant qu'il reste des sommets à sélectionner.

- **Algorithme de Bellman-Ford :**

L'algorithme de Dijkstra ne marche pas toujours quand le graphe contient des arcs dont les coûts sont négatifs. Considérons par exemple le graphe suivant :

Le principe général de l'algorithme consiste à améliorer une arborescence réalisable (initial) (V, A) de racine s à y jusqu'à l'obtention d'une arborescence optimale des plus courts chemins, issue de s si celle-ci existe.

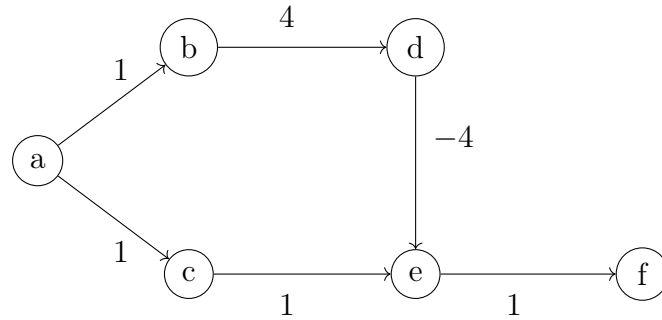


FIGURE 1.10 – Exemple d'un graphe

1.3.2 Le problème du flot maximum

Le flot max est un concept important en théorie des graphes et en théorie des réseaux, qui représente la quantité maximale de flux qu'un réseau donné peut transporter entre une source et un puits.

Un réseau de flot est généralement représenté sous forme de graphe orienté pondéré, où chaque nœud représente une source, un puits ou un nœud intermédiaire, et chaque arc représente une connexion entre deux nœuds avec une capacité qui indique la quantité maximale de flux qui peut être transportée entre eux.

Le flot max peut être calculé à l'aide de l'algorithme de Ford-Fulkerson ou d'autres algorithmes similaires.

1.3.3 Problème d'arbre couvrant

Le problème d'arbre couvrant (également appelé en anglais MST pour Minimum Spanning Tree) est un problème fondamental de la théorie des graphes. Il s'agit de trouver un arbre qui connecte tous les sommets d'un graphe donné, tout en minimisant la somme des poids des arêtes.

Il existe plusieurs algorithmes pour résoudre le problème d'arbre couvrant, tels que l'algorithme de Kruskal. Chacun de ces algorithmes a ses propres avantages et inconvénients en termes de temps d'exécution, de complexité et de performance.

Proposition 1.2. Propriétés d'un arbre couvrant minimal

Un arbre couvrant minimal respecte toujours certaines propriétés. Ces propriétés sont toujours vraies pour tous les MST. Connaître ces propriétés est très important, car de nombreux algorithmes y font référence pour expliquer la logique derrière leur fonctionnement.

1. Si certaines arêtes ont le même poids, il est alors possible qu'il y ait plusieurs MST équivalents en poids pour un graphe G .
2. Si toutes les arêtes ont des poids distincts, alors il n'y a qu'un seul MST.
3. S'il y a un cycle dans le graphe, alors l'arête la plus lourde de ce cycle ne fait pas partie du MST.
4. La coupure d'un graphe connecté est un ensemble d'arêtes qui, une fois enlevées, séparent le graphe en deux composantes disjointes. La propriété de la coupure stipule que la plus petite arête de la coupure fait partie du MST.
5. Si un sommet u possède une arête de poids minimum et unique, cette arête fait toujours partie du MST [10].

- **Algorithme de Kruskal :**

l'algorithme de Kruskal nécessite auparavant le tri des arêtes. Cette condition permet de simplifier grandement le fonctionnement de l'algorithme. Il ne s'agit en fait que de placer les meilleures arêtes les unes après les autres, tant qu'elles ne créent pas de cycle.

Conclusion

En conclusion, la théorie des graphes et l'optimisation sont des outils essentiels dans l'étude des réseaux. La théorie des graphes permet de représenter de manière abstraite les relations entre les différents éléments d'un réseau, tandis que l'optimisation permet de trouver les solutions les plus efficaces pour résoudre les problèmes liés à ces réseaux.

CHAPITRE 2

PROBLÈME DE FLOT À COÛT MINIMUM

Introduction

Dans ce chapitre, nous allons explorer en détail le problème de flot à coût minimum. Nous allons commencer par une présentation théorique du problème, en expliquant les concepts clés tels que les flots, les capacités et les coûts.

Enfin, nous allons appliquer les techniques présentées sur un exemples concrets.

2.1 Problèmes de flots dans les réseaux

Les problèmes de flot sur les réseaux sont des problèmes fondamentaux dans le domaine des réseaux. Ils consistent à acheminer des marchandises à travers un réseau en minimisant les coûts, tout en satisfaisant les demandes spécifiques des terminaux. Ce type de problème permet d'optimiser l'utilisation des ressources disponibles dans le réseau.

Un modèle général de problème de flot est utilisé pour décrire les caractéristiques essentielles, telles que la capacité des arêtes, les demandes des terminaux, les coûts de transfert et les contraintes de conservation du flot. Dans le contexte d'un problème de distribution, le problème de flot à coût minimal est appliqué pour déterminer le chemin optimal afin de minimiser les coûts totaux [14].

2.1.1 Problème de flot général

On se place dans un réseau \mathbf{R} constitué par un ensemble V de noeuds et un ensemble E d'arcs. Ce réseau est modélisé par le **graphe orienté** $G = (V, E)$. Pour un noeud $i \in V$ on note $\Gamma^+(i)$ l'ensemble de ces successeurs, $\Gamma^+(i) = \{j \in V / (i, j) \in E\}$. De même on note, $\Gamma^-(j)$ l'ensemble des prédecesseurs d'un noeud $j \in V$, $\Gamma^-(j) = \{i \in V / (i, j) \in E\}$.

On considère la distribution d'un seul produit sur \mathbf{R} . Chaque noeud de ce réseau est étiqueté par b_i , la quantité de produit disponible au noeud $i \in V$. L'ensemble des noeuds V est partitionné en trois sous-ensembles S, T et D :

S est l'ensemble des noeuds sources (usines, production) avec $b_i > 0$ pour $i \in S$,

T est l'ensemble des noeuds de transfert ou transbordement avec $b_i = 0$ pour $i \in T$,

D est l'ensemble des noeuds destinations (client, demandes) $b_i < 0$ pour $i \in D$

On suppose que les offres et les demandes sont équilibrées, i.e

$$\sum_{i \in S \cup T \cup D} b_i = 0.$$

Les arcs du réseau peuvent avoir une capacité u_{ij} . On note x_{ij} les variables de flots.

Le problème de "flot" du réseau de distribution, se modélise sous la forme du programme mathématique décrit en bas :

$$\min \sum_{ij \in E} c_{ij}^T x_{ij} \tag{2.1}$$

$$s.c. \sum_{j \in \Gamma^+(i)} x_{ij} - \sum_{j \in \Gamma^-(i)} x_{ji} = b_i, \forall i \in V \tag{2.2}$$

$$0 \leq x_{ij} \leq u_{ij}, \forall (i, j) \in E \tag{2.3}$$

La contrainte (2.2) est une écriture compacte de la contrainte de conservation du flot au niveau des noeuds du réseau [21].

2.1.2 Problème de flot à coût minimal

Soit $G = (V, E, u, c)$ un réseau de distribution, avec $n = |V|$ et $m = |E|$ représentant respectivement le nombre de noeuds et d'arcs de G , $u = (u_{ij}, (i, j) \in E)$ est le vecteur des capacités, $c = (c_{ij}, (i, j) \in E)$ est le vecteur des coûts de distribution sur les arcs du réseau [2].

Dans un problème de flux à coût minimal, une solution est définie en spécifiant le flux x_{ij}

dans chaque arc (i, j) du réseau. Une solution peut alors être représentée par une matrice x ayant m composantes. Une solution est réalisable si et seulement si [20] :

- 1- pour chaque nœud de transit, le flux total entrant est égal au flux total sortant.
- 2- pour chaque puits, le flux total entrant est égal à la demande du nœud plus le flux total sortant.
- 3- Pour chaque source, le flux sortant total est égal à l'offre du nœud plus le flux entrant total.
- 4- Tous les flux d'arcs sont non négatifs.

Les conditions (1)-(4) sont englobées par la contrainte (2.5)

Le problème consiste à déterminer un vecteur de débit x qui minimise le coût $Z = c^T x$ parmi toutes les solutions possibles. Ces conditions, ainsi que la fonction objectif du problème, peuvent être exprimées comme suit :

$$\min Z = \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (2.4)$$

$$s.c. \sum_{j \in \Gamma^+(i)} x_{ij} - \sum_{j \in \Gamma^-(i)} x_{ij} = \begin{cases} f, & i = s, \\ -f, & i = p, \\ 0, & \text{sinon, } \forall i = 1, \dots, n \end{cases} \quad (2.5)$$

$$0 \leq x_{ij} \leq u_{ij}, \quad \forall (i, j) \in E. \quad (2.6)$$

2.2 Méthode de simplexe réseau

La méthode du simplexe réseau est une extension de la méthode du simplexe standard, qui est spécialement conçue pour résoudre les problèmes de flot à coût minimum comme les problèmes de distribution dans lesquels il y a des contraintes de capacité et de coût associées à un réseau de distribution [20]. Dans cette section, nous décrivons en détail l'algorithme du simplexe réseau. Nous montrons ensuite comment calculer les flux d'arcs et les potentiels de nœuds pour n'importe quelle structure de base. Nous expliquons ensuite comment effectuer diverses opérations du simplexe telles que la sélection des arcs entrants, des arcs sortants et des pivots à l'aide de la structure de données de l'arbre.

2.2.1 Définition et concepts de base

Définition 2.1. Soit x un flot et T un arbre couvrant, la paire (x, T) est dite solution basique si elle vérifie $Ax = b$ et $x_{ij} = 0$ ou $x_{ij} = u_{ij}$ [2].

Définition 2.2. $\{x, T\}$ SRS est dite non dégénérée si elle vérifie :

$$0 < x_{ij} < u_{ij}, \forall (i, j) \in T \text{ [3].}$$

Lemme 2.1. Soit $T \subset A$ un ensemble de colonnes de A tel que les arcs correspondants forment un arbre couvrant. Les colonnes de T sont donc linéairement indépendantes [20].

Vecteurs des coûts réduits

$\delta = c - A^T y$, avec $\delta \in \mathbb{R}^n$ et $y \in \mathbb{R}^n$ est appelé vecteur des potentiels.

Pour chaque arc $(i, j) \in E$, on aura : $\delta_{ij} = c_{ij} - y_i + y_j$ [2].

On pose :

$$\delta_{ij} = 0, \quad \forall (i, j) \in T \tag{2.7}$$

$$\delta_{ij} = c_{ij} - y_i + y_j \quad \forall (i, j) \in T$$

Critère d'optimalité

Soit x une solution réalisable basique du problème PFCM et T l'arbre associé. on a :

$$\begin{cases} \delta_{ij} \geq 0 & \text{pour } x_{ij} = 0 \\ \delta_{ij} \leq 0 & \text{pour } x_{ij} = u_{ij}, \quad (i, j) \in T \end{cases} \tag{2.8}$$

Alors x est une solution optimale du PFCM [22].

Choix de la variable entrante :

Dans le cas où les relations (2.7) ne sont pas vérifiées, on choisit une variable $x_{i_0j_0}$, alors sa valeur augmente et diminue si :

$$x_{i_0j_0} = u_{i_0j_0} \text{ ou } x_{i_0j_0} = 0$$

Une variable entre en base si on a :

$$\begin{cases} x_{ij} = 0 \text{ et } \delta_{ij} < 0 \Rightarrow \text{on augmente le flot } x_{ij} \text{ sur l'arc } (i, j), \\ x_{ij} = u_{ij} \text{ et } \delta_{ij} > 0 \Rightarrow \text{on baisse le flot } x_{ij} \text{ sur l'arc } (i, j), \end{cases}$$

Calculons :

$$\begin{cases} \varepsilon_1 = \min_{(i,j) \notin T} \delta_{ij}, & \{x_{ij} = 0 \text{ et } \delta_{ij} < 0\}, \\ \varepsilon_2 = \max_{(i,j) \notin T} \delta_{ij}, & \{x_{ij} = u_{ij} \text{ et } \delta_{ij} > 0\}, \end{cases}$$

Soit $\varepsilon = \min\{\varepsilon_1, \varepsilon_2\}$ et on a 2 cas :

Si $\varepsilon = \varepsilon_1 = \delta_{i_0 j_0} < 0$, alors $x_{i_0 j_0}$ doit augmenter.

Si $\varepsilon = \varepsilon_2 = \delta_{i_0 j_0} > 0$, alors $x_{i_0 j_0}$ doit diminuer.

L'ajout d'un arc (i_0, j_0) à l'arbre T crée un cycle élémentaire φ , où le flot sur les arcs de φ change et pour les autres arcs ne changent pas.

- Si on a $x_{i_0 j_0} = 0$, alors on doit augmenter la valeur du flot des arcs de φ ayant le même sens que $(x_{i_0 j_0})$ noté φ^+ de θ unités et le diminuer de θ unités pour les arcs ayant un sens de parcours opposé noté φ^- .
- Par contre, si $x_{i_0 j_0} = u_{i_0 j_0}$ alors on fait l'inverse, on diminue la quantité du flot de θ unités pour les arcs de φ^+ et on l'augmente pour les arcs de φ^- .
- Le changement maximum de la valeur du flot noté θ , est calculé comme suit :

$$\theta = \min\{\theta^-, \theta^+\}$$

avec :

$$\begin{cases} \theta^- = \min_{(i,j) \in \varphi^-} \{x_{ij}\} \\ \theta^+ = \min_{(i,j) \in \varphi^+} \{u_{ij} - x_{ij}\} \end{cases} \quad (2.9)$$

La valeur du nouveau flot :

$$\begin{cases} x_{ij} + \theta, & (i, j) \in \varphi^+ \\ x_{ij} - \theta, & (i, j) \in \varphi^- \\ x_{ij}, & \text{sinon} \end{cases} \quad (2.10)$$

- Changement de base :

En envoyant une quantité supplémentaire de flot de θ unités autour du cycle φ , alors il y a une variable basique notée x_{i_1, j_1} qui atteindra l'une de ses bornes (0 ou u_{i_1, j_1}). Par conséquent, l'arc (i_1, j_1) est supprimé de la base (arbre T) selon les cas suivants :

- Si $x_{i_1, j_1} \in T$ alors on pose, $T = T \cup (i_0, j_0) \setminus (i_1, j_1)$.
- Si l'arc $(i_1, j_1) = (i_0, j_0) \notin T$, alors on pose $T = T$.

Pour calculer les multiplicateur y_i , il suffit de fixer la valeur d'un multiplicateur (par exemple $y_t = 0$) et de calculer séquentielle-ment les autres inconnus [21].

Lemme 2.2.

- Si les demandes et les offres $b_i, i \in V$. Ainsi que les capacités $u_{ij} \in E$ sont toutes entières, alors il existe une solution optimale primale x qui est entière.
- Si de plus les coûts c_{ij} sont tous entières, alors il existe une solution optimale y entière de son problème dual associé [2].

2.3 Exemple pratique

Exemple 2.1. Soit $G = (V, E, u, c)$ un réseau de transport, avec $n = |V| = 6$ et $m = |E| = 8$ représentant respectivement le nombre de noeuds et d'arcs de G telque :

$u = (u_{ij}, (i, j) \in E)$ est le vecteur des capacités.

$c = (c_{ij}, (i, j) \in E)$ est le vecteur des coûts de transport sur les arcs du réseau.

$b = (b_i, i \in V)$ est le vecteur des demandes.

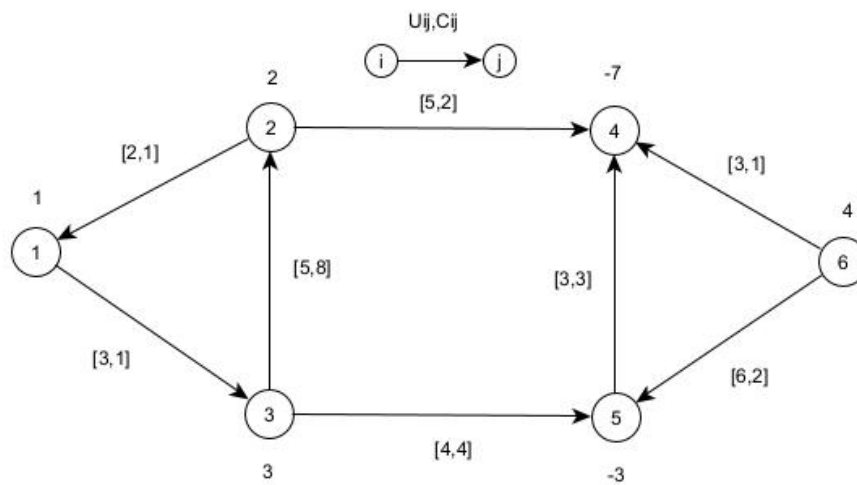


FIGURE 2.1 – Réseau de transport

$$\text{Min}Z = x_{21} + x_{13} + 8x_{32} + 2x_{24} + 4x_{35} + 3x_{54} + x_{64} + 2x_{65}$$

$$S.c \left\{ \begin{array}{l} +x_{13} - x_{21} = 1 \\ +x_{21} - x_{32} + x_{24} = 2 \\ -x_{13} + x_{32} + x_{35} = 3 \\ -x_{24} - x_{54} - x_{64} = -7 \\ -x_{35} + x_{54} - x_{65} = -3 \\ x_{64} + x_{65} = 4 \end{array} \right.$$

$$0 \leq x_{21} \leq 2; 0 \leq x_{32}, x_{24} \leq 5; 0 \leq x_{35} \leq 4; 0 \leq x_{54} \leq 3; 0 \leq x_{64} \leq 3 \quad 0 \leq x_{13}, x_{65} \leq 6$$

Une solution réalisable : $x_{13} = 2; x_{21} = 1; x_{32} = 2; x_{24} = 3; x_{35} = 3; x_{54} = 3;$
 $x_{64} = 1; x_{65} = 3$

Considérons l'arbre couvrant $T = (2, 1); (3, 2); (2, 4); (3, 5); (6, 5)$ la solution basique correspondante.

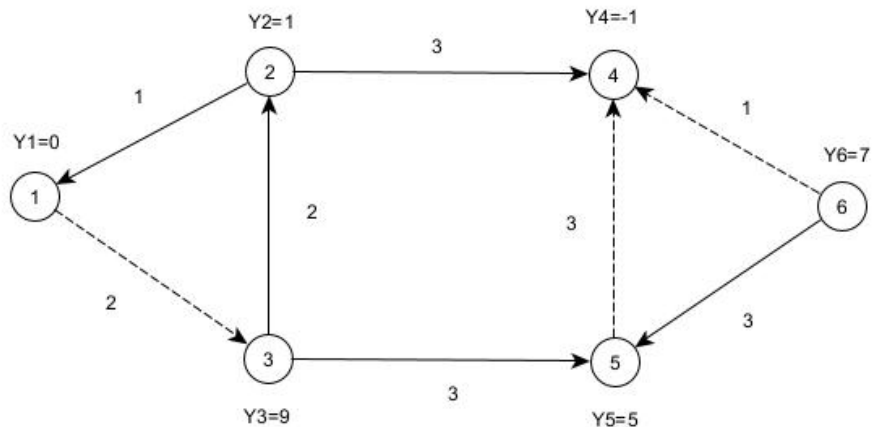


FIGURE 2.2 – Choix de l'arbre couvrant

• **Iteration 1** Calcul des vecteurs des coûts réduits basiques :

on pose $y_1 = 0$ avec $\delta_i = 0$

$$\delta_{21} = c_{21} - y_2 + y_1 \Rightarrow y_2 = 1$$

$$\delta_{32} = c_{32} - y_3 + y_2 \Rightarrow y_3 = 9$$

$$\delta_{35} = c_{35} - y_3 + y_5 \Rightarrow y_5 = 5$$

$$\delta_{65} = c_{65} - y_6 + y_5 \Rightarrow y_6 = 7$$

$$\delta_{24} = c_{24} - y_2 + y_4 \Rightarrow y_4 = -1$$

Calcul des vecteurs des coûts réduits non basiques :

$$\delta_{13} = c_{13} - y_1 + y_3 = 1 - 0 + 9 = 10 > 0 \text{ et } x_{13} = 2 \text{ Nopt}$$

$$\delta_{54} = c_{54} - y_5 + y_4 = 3 - 5 - 1 = -3 < 0 \text{ et } x_{54} = 3 = u_{54} \text{ opt}$$

$$\delta_{64} = c_{64} - y_6 + y_4 = 1 - 7 - 1 = -7 < 0 \text{ et } x_{64} = 1 \text{ Nopt}$$

• **Variable Entrante**

$$\delta_{ij} = \max_{(i,j) \in T_{\text{Nopt}}} \{|\delta_{ij}|\}, \text{ avec } T_{\text{Nopt}} = \{(i,j) \notin T : (x_{ij} = 0 \wedge \delta_{ij} < 0) \vee (x_{ij} = u_{ij} \wedge \delta_{ij} > 0)\}$$

$$\delta_{ij} = \max(10, |-7|) = 10$$

Alors, on fait entrer l'arc (1, 3) en base pour former le cycle φ suivant :

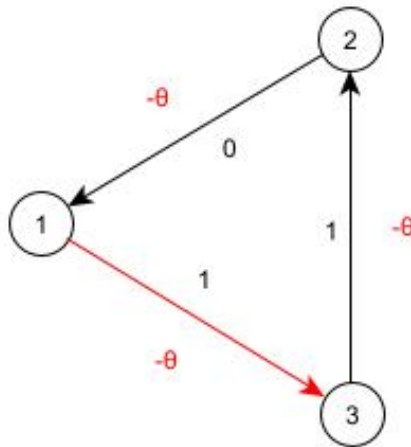


FIGURE 2.3 – Choix de la variable entrante

$$- \theta^- = \min_{(i,j) \in \varphi^-} \{x_{ij}\} = \min\{x_{13}, x_{32}, x_{21}\} = \min\{2, 2, 1\} = 1$$

$$- \theta^+ = \min_{(i,j) \in \varphi^+} \{u_{ij} - x_{ij}\} = \emptyset$$

$$- \theta = \min\{\theta^+, \theta^-\} = 1$$

• **Variable sortante**

L'arc (2, 1) sort de la base, donc la base devient $T = T \cup (1, 3) \setminus (2, 1)$.

$$T = \{(1, 3); (3, 2); (2, 4); (3, 5); (6, 5)\}$$

• **Iteration 2**

On vas vérifier les conditions d'optimalité pour le nouveau marquage

Calcul des vecteurs des coûts réduits basiques :

$$- \delta_{13} = c_{13} - y_1 + y_3 = 1 - y_1 + y_3 \Rightarrow y_3 = -1$$

$$- \delta_{32} = c_{32} - y_3 + y_2 = 8 + 1 + y_2 \Rightarrow y_2 = -9$$

$$\text{--- } \delta_{24} = c_{24} - y_2 + y_4 = 2 + 9 + y_4 \Rightarrow y_4 = -11$$

$$\text{--- } \delta_{35} = c_{35} - y_3 + y_5 = 4 + 1 + y_5 \Rightarrow y_5 = -5$$

$$\text{--- } \delta_{65} = c_{65} - y_6 + y_5 = 2 - y_6 - 5 \Rightarrow y_6 = -3$$

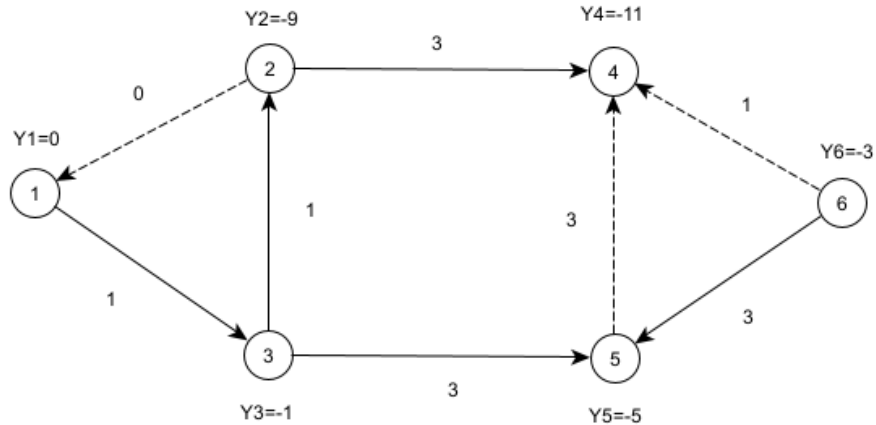


FIGURE 2.4 – Le nouveau marquage

Calcul des vecteurs des coûts réduits non basiques :

$$\text{--- } \delta_{21} = c_{21} - y_2 + y_1 = 1 + 9 + 0 = 10 \text{ et } x_{21} = 0 \text{ opt}$$

$$\text{--- } \delta_{54} = c_{54} - y_5 + y_4 = 3 + 2 - 8 = -3 < 0 \text{ et } x_{54} = 3 = u_{54} \text{ opt}$$

$$\text{--- } \delta_{64} = C_{64} - y_6 + y_4 = -7 < 0 \text{ et } x_{64} = 1 \text{ Nopt}$$

• **Variable Entrante**

Il n'ya pas qu'un seul arc non optimal qui est l'arc (6,4) donc c'est l'arc entrant.

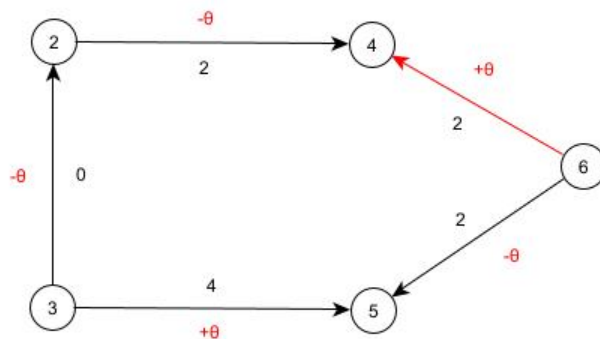


FIGURE 2.5 – Choix de la variable entrante (itération 2)

$$\text{--- } \theta^- = \min_{(i,j) \in \varphi^-} \{x_{ij}\} = \min\{x_{65}, x_{32}, x_{24}\} = \min\{3, 1, 3\} = 1$$

$$\text{--- } \theta^+ = \min_{(i,j) \in \varphi^+} \{u_{ij} - x_{ij}\} = \min\{u_{64} - x_{64}, u_{35} - x_{35}\} = \min\{3 - 1; 4 - 3\} = 1$$

$$\text{--- } \theta = \min\{1, 1\} = 1$$

• **Variable sortante**

L'arc (3, 2) sort de la base, donc la base devient $T = T \cup (6, 4) \setminus (3, 2)$.

$$T = \{(1, 3); (2, 4); (3, 5); (6, 5); (6, 4)\}$$

• **Iteration 3**

Calcul des vecteurs des coûts réduits basiques :

$$\text{--- } \delta_{13} = c_{13} - y_1 + y_3 = 1 - 0 + y_3 \Rightarrow y_3 = -1$$

$$\text{--- } \delta_{24} = c_{24} - y_2 + y_4 = 2 - y_2 + 4 \Rightarrow y_2 = -2$$

$$\text{--- } \delta_{35} = c_{35} - y_3 + y_5 = 4 + 1 + y_5 \Rightarrow y_5 = -5$$

$$\text{--- } \delta_{64} = c_{64} - y_6 + y_4 = 1 + 3 + y_4 \Rightarrow y_4 = -4$$

$$\text{--- } \delta_{65} = c_{65} - y_6 + y_5 = 2 - y_6 - 5 \Rightarrow y_6 = -3$$

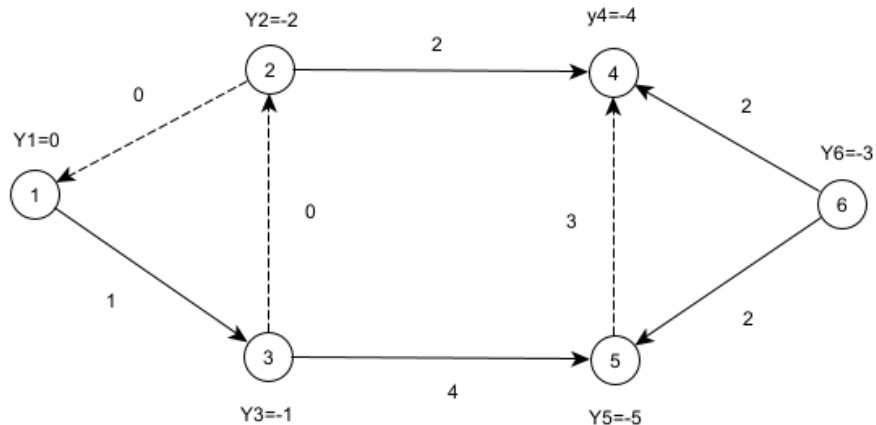


FIGURE 2.6 – Mise à jour des potentiels

Calcul des vecteurs des coûts réduits non basiques :

$$\text{--- } \delta_{21} = 10 \text{ et } x_{21} = 0 \text{ opt}$$

$$\text{--- } \delta_{54} = -8 < 0 \text{ et } x_{54} = 3 = u_{54} \text{ opt}$$

$$\text{--- } \delta_{64} = -7 < 0 \text{ et } x_{64} = 1 \text{ Nopt}$$

- **Variable Entrante**

Il n'ya pas qu'un seul arc non optimal qui est l'arc (5,4) donc c'est l'arc entrant.

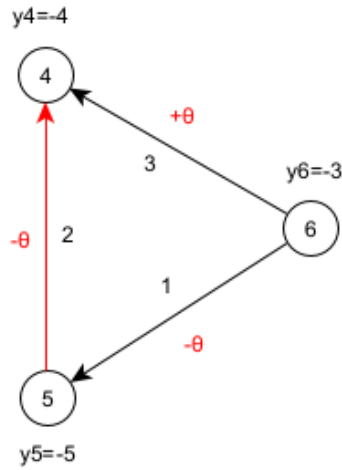


FIGURE 2.7 – Choix de la variable entrante (itération 3)

- $\theta^- = \min_{(i,j) \in \varphi^-} \{x_{ij}\} = \min\{x_{65}, x_{54}\} = 2$
- $\theta^+ = \min_{(i,j) \in \varphi^+} \{u_{ij} - x_{ij}\} = \min\{u_{64} - x_{64}\} = 1$
- $\theta = \min\{2, 1\} = 1$

- **Variable sortante**

L'arc (6, 4) sort de la base, donc la base devient $T = T \cup (5, 4) \setminus (6, 4)$.

$$T = \{(1, 3); (2, 4); (3, 5); (6, 5); (5, 4)\}$$

- **Iteration 4**

Calcul des vecteurs des coûts réduits basiques :

- $\delta_{13} = c_{13} - y_1 + y_3 = 1 - 0 + y_3 \Rightarrow y_3 = -1$
- $\delta_{24} = c_{24} - y_2 + y_4 = 2 - y_2 - 8 \Rightarrow y_2 = -6$
- $\delta_{35} = c_{35} - y_3 + y_5 = 4 + 1 + y_5 \Rightarrow y_5 = -5$
- $\delta_{54} = c_{54} - y_5 + y_4 = 1 + 3 + y_4 \Rightarrow y_4 = -8$
- $\delta_{64} = c_{64} - y_6 + y_4 = 1 - y_6 - 8 \Rightarrow y_6 = -7$

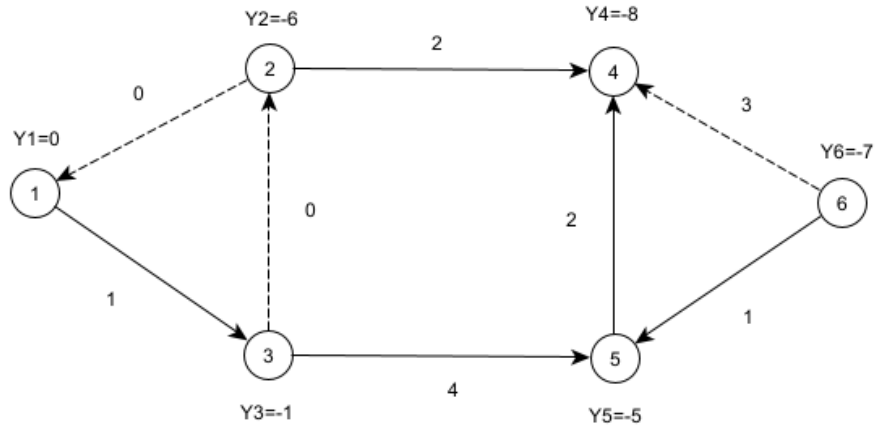


FIGURE 2.8 – La solution optimal

Calcul des vecteurs des coûts réduits non basiques :

- $\delta_{21} = 7$ et $x_{21} = 0$ opt
- $\delta_{32} = 3$ et $x_{32} = 0$ opt
- $\delta_{64} = 0$ et $x_{64} = 3 = u_{64}$ opt

Par conséquent la solution optimal est :

$$x^* = (x_{21} = 0; x_{13} = 1; x_{32} = 0; x_{35} = 4; x_{54} = 2; x_{64} = 3; x_{65} = 1; x_{24} = 2)$$

Le coût du flot

$$Z^*(x) = 1 \times 0 + 1 \times 1 + 8 \times 0 + 2 \times 2 + 4 \times 4 + 3 \times 2 + 1 \times 3 + 2 \times 1$$

$$Z^*(x) = 32$$

Conclusion

Dans ce chapitre, nous avons étudié le problème de flot à coût minimum qui est un problème fondamental en optimisation dans les réseaux. Ce problème consiste à déterminer le flot optimal à travers les arcs d'un réseau afin de minimiser le coût total de transport. Ce dernier trouve son application dans divers domaines, tels que la logistique et le transport. .

Introduction

La méthode du support est souvent utilisée en conjonction avec la méthode du simplexe standard ou la méthode du simplexe réseau pour résoudre des problèmes de programmation linéaire. Elle peut aider les entreprises à mieux comprendre les implications de leurs choix de planification et à prendre des décisions plus éclairées pour optimiser leur processus de production ou de distribution.

La méthode du support peut également être utilisée pour évaluer la sensibilité des solutions optimales du problème primal aux variations des paramètres du problème, tels que les coûts des ressources ou les exigences de production. Cela permet aux entreprises de mieux gérer les risques associés à leur chaîne d'approvisionnement et d'optimiser leur stratégie de production ou de distribution en conséquence.

3.1 Position du problème et définitions

Le problème standard de la programmation linéaire s'écrit sous la forme suivante :

$$\max Z = c^T x, \quad (3.1)$$

$$Ax = b, \quad (3.2)$$

$$x \geq 0, \quad (3.3)$$

où c et x sont des n -vecteurs, b est un m -vecteur ; $A = A(I, J)$ est une matrice d'ordre $m \times n$, avec $\text{rang}A = m < n$. La fonction Z est dite fonction objectif ; les équations (3.2) sont les contraintes principales, tandis que les contraintes (3.3) sont dites simples [4]. Notons par $I = \{1, 2, \dots, m\}$ et $J = \{1, 2, \dots, n\}$ les ensembles d'indices des lignes et des colonnes de A .

Un vecteur x vérifiant les contraintes (3.2) et (3.3) est appelé solution réalisable du problème (3.1)- (3.3). Une solution réalisable x^0 est dite optimale si :

$$Z(x^0) = c^T x^0 = \max_x c^T x,$$

où x est pris parmi toutes les solutions réalisables du problème (3.1)-(3.3).

D'autre part, une solution réalisable x^ε est appelée ε -optimale (ou suboptimale) si :

$$0 \leq Z(x^0) - Z(x^\varepsilon) = c^T x^0 - c^T x^\varepsilon \leq \varepsilon,$$

où x^0 est une solution optimale du problème (3.1)-(3.3) et ε est un nombre positif ou nul choisi comme précision.

Soit un sous-ensemble d'indices $J_B \in J$, tel que $|J_B|=|I| = m$. L'ensemble J_B est alors appelé support si :

$$\det A_B = \det A(I, J_B) \neq 0.$$

Le couple $\{x, J_B\}$ formé de la solution réalisable x et du support J_B est appelé Solution Réalisable de Support (SRS). La solution réalisable de support est dite non dégénérée si :

$$x_j > 0, \forall j \in J_B. \quad (3.4)$$

3.1.1 Formule d'accroissement de la fonction objectif

Soit $\{x, J_B\}$ une solution réalisable de support du problème (3.1)-(3.3). Afin de calculer l'accroissement de la fonction objectif, considérons une autre solution réalisable $\bar{x} = x + \Delta x$. L'accroissement de la fonction objectif s'écrit alors :

$$\Delta Z = Z(\bar{x}) - Z(x) = c^T \bar{x} - c^T x = c^T \Delta x \quad (3.5)$$

Par ailleurs, on a :

$$A\bar{x} = Ax = b \Rightarrow A\Delta x = 0,$$

En posant :

$$\Delta x = \begin{pmatrix} \Delta x_B \\ \Delta x_N \end{pmatrix}, \quad \Delta x_B = \Delta x(J_B), \quad \Delta x_N = \Delta x(J_N),$$

l'égalité $A\Delta x = 0$ peut ainsi s'écrire :

$$A_B\Delta x_B + A_N\Delta x_N = 0 \Rightarrow \Delta x_B = -A^{-1}_B A_N\Delta x_N, \quad (3.6)$$

En vertu de cette relation, l'accroissement **(3.5)** devient :

$$\begin{aligned} Z(\bar{x}) - Z(x) &= \Delta Z = c^T \Delta x \\ &= c_B^T \Delta x_B + c_N^T \Delta x_N \\ &= -(c_B^T A^{-1}_B A_N - c_N^T) \Delta x_N. \end{aligned} \quad (3.7)$$

On définit le vecteur des potentiels u :

$$u^T = c_B^T A^{-1}_B,$$

ainsi que le vecteur des coûts réduits E , appelé aussi vecteur des estimations :

$$E^T = E^T(J) = u^T A - c^T = (E_B^T, E_N^T),$$

où

$$\begin{aligned} E_B^T &= u^T A_B - c_B^T = c_B^T A^{-1}_B A_B - c_B^T = 0, \\ E_N^T &= u^T A_N - c_N^T \Leftrightarrow E_j = u^T a_j - c_j, j \in J_N, \end{aligned}$$

Par conséquent, l'accroissement **(3.7)** prend la forme finale suivante :

$$\Delta Z = Z(\bar{x}) - Z(x) = -E_N^T \Delta x_N = - \sum_{j \in J_N} E_j (\bar{x}_j - x_j). \quad (3.8)$$

3.1.2 Critère d'optimalité

Théorème 3.1 ([13]). *Soit $\{x, J_B\}$ une solution réalisable de support du problème (3.1)-(3.3).*

Alors les relations suivantes :

$$\begin{cases} E_j \geq 0, \text{ si } x_j = 0, \\ E_j = 0, \text{ si } x_j > 0, \forall j \in J_N, \end{cases} \quad (3.9)$$

sont suffisantes, et dans le cas de la non-dégénérescence aussi nécessaires, pour l'optimalité de la solution réalisable x .

Démonstration. voir [4]

3.1.3 Estimation de suboptimalité

Pour estimer l'écart qui existe entre la valeur optimale $Z(x^0)$ et une autre valeur $Z(x)$ d'une SRS $\{x, J_B\}$, on remplace dans la formule d'accroissement (3.8) le vecteur \bar{x} par x^0 et on aura :

$$Z(x^0) - Z(x) = - \sum_{j \in J_N} E_j(x_j^0 - x_j) = \sum_{j \in J_N} E_j(x_j - x_j^0).$$

Supposons que $E_N \geq 0$. Puisque $x^0 \geq 0$, on peut alors écrire :

$$x_j - x_j^0 \leq x_j \Rightarrow E_j(x_j - x_j^0) \leq E_j x_j,$$

d'où

$$0 \leq Z(x^0) - Z(x) \leq \sum_{j \in J_N} E_j x_j. \quad (3.10)$$

Pour $E_N \geq 0$, le nombre :

$$\beta(x, J_B) = \sum_{j \in J_N} E_j x_j.$$

est ainsi appelé estimation de suboptimalité.

3.1.4 Algorithme de résolution

Étant donné un nombre réel quelconque $\varepsilon \geq 0$, choisi comme précision, et $\{x, J_B\}$ une SRS initiale du problème (3.1)-(3.3), le but de l'algorithme est alors de construire une solution réalisable ε -optimale x^ε ou carrément une solution optimale x^0 . L'itération de l'algorithme consiste à passer de $\{x, J_B\}$ vers une autre SRS $\{\bar{x}, \bar{J}_B\}$ telle que $Z(\bar{x}) \geq Z(x)$. A cet effet, on construit la nouvelle solution réalisable \bar{x} de la manière suivante :

$$\bar{x} = x + \theta l, \quad \theta \geq 0,$$

où l est un n -vecteur appelé direction d'amélioration, θ étant le pas le long de cette direction. Dans cet algorithme, on choisit la métrique du simplexe, qui ne fait varier qu'une seule composante parmi toutes celles qui ne vérifient pas les relations (3.9). Pour que l'accroissement :

$$Z(\bar{x}) - Z(x) = \theta |E_{j_0}| > 0,$$

soit maximal, il faut alors prendre θ aussi grand que possible et choisir l'indice j_0 tel que :

$$|E_{j_0}| = \max_{j \in J_{N \setminus O}} |E_j|,$$

où $J_{NNO} \subset J_N$ est l'ensemble des indices non optimaux, c'est-à-dire l'ensemble des indices qui ne vérifient pas les relations d'optimalité, à savoir

$$J_{NNO} = \{j \in J_N : [x_j = 0 \text{ et } E_j < 0] \vee [x_j > 0 \text{ et } E_j \neq 0]\}$$

On pose donc :

$$\begin{cases} l_{j_0} = -\text{sign } E_{j_0}, \\ l_j = 0, j \neq j_0, j \in J_N, \\ l_B = A^{-1}a_{j_0} \text{ sign } E_{j_0}. \end{cases} \quad (3.11)$$

D'autre part, le pas θ doit vérifier les inégalités suivantes :

$$x_j + \theta l_j \geq 0, j \in J_B, \quad (3.12)$$

$$x_{j_0} + \theta l_{j_0} \geq 0, \quad (3.13)$$

En calculant les différentes valeurs maximales que peut prendre θ dans les relations **(3.12)**, on trouve :

$$\theta_j = \begin{cases} \frac{-x_j}{l_j}, \text{ si } l_j < 0, \\ \infty, \text{ si } l_j \geq 0. \end{cases}$$

D'où :

$$\theta_{j_1} = \min_{j \in J_B} \theta_j. \quad (3.14)$$

Pour l'indice j_0 , la relation **(3.13)** donne :

$$\theta_{j_0} = \begin{cases} x_{j_0}, \text{ si } l_{j_0} = -1, \\ \infty, \text{ si } l_{j_0} = 1. \end{cases} \quad (3.15)$$

Par conséquent, le pas maximal θ^0 le long de la direction l est égal à :

$$\theta^0 = \min\{\theta_{j_1}, \theta_{j_0}\}, \quad (3.16)$$

- Si $\theta^0 = \infty$, alors le problème **(3.1)-(3.3)** est non borné.
- Sinon, en tenant compte des formules **(3.11)** et **(3.16)**, la nouvelle solution réalisable \bar{x} s'écrit : donc :

$$\bar{x} = x + \theta^0 l, Z(\bar{x}) = Z(x) + \theta^0 |E_{j_0}|,$$

Pour $E_N \geq 0$, calculons l'estimation de suboptimalité de la nouvelle SRS $\{\bar{x}, J_B\}$:

$$\beta(\bar{x}, J_B) = \sum_{j \in J_N} E_j \bar{x}_j,$$

Les composantes \bar{x}_j , $j \in J_N$, s'écrivent :

$$\begin{cases} \bar{x}_j = x_j, j \neq j_0, j \in J_N, \\ \bar{x}_{j_0} = x_{j_0} - \theta^0. \end{cases}$$

D'où :

$$\beta(\bar{x}, J_B) = \beta(x, J_B) - \theta^0 E_{j_0}.$$

- Si $\beta(x, \bar{J}_B \leq \varepsilon)$, alors la solution réalisable \bar{x} est ε -optimale et on peut arrêter l'algorithme.
- Sinon, on change le support J_B de la manière suivant :

1. Si $\theta^0 = \theta_{j_0}$, alors l'indice j_0 est devenue optimal et on ne change pas de support. On écrira donc :

$$\bar{x} = x + \theta^0 l, \bar{J}_B = J_B,$$

.

2. Si $\theta^0 = \theta_{j_1}$, alors la composante l_{j_1} est négative et donc forcément non nulle. Dans ce cas, la composante \bar{x}_{j_1} est devenue critique ($\bar{x}_{j_1} = 0$). Pour éviter cette dégénérescence, il faut alors faire sortir j_1 de J_B et le remplacer par j_0 . On doit prouver que le nouvel ensemble est un support. En effet, on a :

$$l_{j_1} \neq 0 = e_j^T A_B^{-1} a_{j_0} \text{sign } E_{j_0} = x_{j_1 j_0} \text{sign } E_{j_0} \neq 0,$$

où $A_B^{-1} a_{j_0} = X(J_B) = (x_{j j_0}, j \in J_B)$, et e_{j_1} représente le vecteur unitaire dont la composante j_1 vaut 1.

Nous avons donc $x_{j_1 j_0} \neq 0$ et montrons alors que l'ensemble $\bar{J}_B = (J_B \setminus j_1) \cup j_0$ est un support, c'est-à-dire :

$$\det \bar{A}_B + \det A(I, \bar{J}_B) \neq 0,$$

En effet, de la relation $A_B^{-1} a_{j_0} = X(J_B)$, on déduit le système linéaire suivant :

$$A_B X(J_B) = a_{j_0},$$

où $X(J_B)$ est le vecteur des inconnues. En utilisant la règle de Cramer, la variable $x_{j_1 j_0}$ vaut donc :

$$x_{j_1 j_0} = \frac{\det \bar{A}_B}{\det A_B} \neq 0 \Rightarrow \det \bar{A}_B \neq 0.$$

Par conséquent, \bar{J}_B est un support et la nouvelle SRS $\{\bar{x}, \bar{J}_B\}$ s'écrit :

$$\bar{x} = x + \theta^0 l, \bar{J}_B = (J_B \setminus j_1) \cup j_0.$$

Si la SRS $\{\bar{x}, \bar{J}_B\}$ ne satisfait pas le critère d'optimalité ou celui de suboptimalité, on recommence alors une nouvelle itération à partir de la nouvelle SRS obtenue, et ce, en posant $x := \bar{x}$ et $J_B := \bar{J}_B$.

3.2 Algorithme de support

Algorithm 1 Algorithme de support

. Soit $\{x, J_B\}$, une solution réalisable de support initiale du problème (3.1)-(3.3) et ε un nombre positif ou nul, choisi comme précision.

• (01). Calculer $u^T = c_B^T A_B^{-1}$ et $E_j = u^T a_j - c_j, j \in J_N$

• (02). Test d'optimalité de la SRS $\{x, J_B\}$:

Cas 1 : $E_N \geq 0$.

- Calculer l'estimation de suboptimalité $\beta(x, J_B) = E_N^T x_N$.

si $(\beta(x, J_B) \leq \varepsilon)$ **alors**

| le processus de résolution s'arrête avec la SRS ε -optimale (x, J_B)

sinon

| Aller à (3).

fin

Cas 2 : $E_N \not\geq 0$. Allez à l'étape 3

• (03). Amélioration de la solution réalisable x .

- Déterminer l'ensemble des indices non optimaux :

$$J_{NNO} = \{j \in J_N : E_j \leq 0 \cup [E_j \geq \text{et } x_j \geq 0]\}$$

- Choisir l'indice j_0 telque $|E_{j_0}| = \max_{j \in J_{NNO}} |E_j|$;

- Calculer la direction d'amélioration l en utilisant les relations (3.11) ;

- Calculer le pas $\theta^0 = \min\{\theta_{j_1}, \theta_{j_0}$ où θ_{j_1} et θ_{j_0} son donner par les relations (3.14) et (3.15) ;

si $(\theta^0 = \infty)$ **alors**

| Le problème est non borné et l'algorithme est arrêté ;

sinon

| calculer $\bar{x} = x + \theta^0 l, Z(\bar{x}) = Z(x) + \theta^0 |E_{j_0}|$ Aller à (4).

fin

• (04). Test d'optimalité de la nouvelle SRS \bar{x} :

Algorithm 2 Suite d'algorithme de support

Cas 1 : $E_N \geq 0$.

-Calculer $\beta(\bar{x}, J_B) = \beta(x, J_B) - \theta^0 E_{j_0}$

si $(\beta(\bar{x}, J_B)) < \epsilon$ **alors**

| la SRS $\{\bar{x}, J_B\}$ est ϵ -optimale et l'algorithme est arrêté ;

sinon

| Aller à (5).

fin

Cas 2 : $E_N \not\geq 0$.

• (05). Changement de support.

–Si $\theta^0 = \theta_{j_0}$, $\bar{J}_B = J_B$;

–Si $\theta^0 = \theta_{j_1}$, alors $\bar{J}_B = (J_B \setminus j_1) \cup j_0$;

On pose $x := \bar{x}$ et $J_B := \bar{J}_B$. Aller à (1).

Conclusion

En conclusion, la méthode directe du support est une approche efficace et pratique pour résoudre des problèmes de support et d'optimisation dans divers domaines. Au cours de ce chapitre, nous avons exploré les principes fondamentaux de la méthode directe du support et en continuant à explorer et à développer cette méthode dans le chapitre qui suit pour le cas de notre thème.

CHAPITRE 4

APPLICATION DE LA MÉTHODE DE SUPPORT SUR LES RÉSEAUX

Introduction

La théorie des réseaux fournit des outils mathématiques et informatiques pour analyser et résoudre des problèmes complexes associés à ces réseaux.

Dans ce chapitre nous nous intéressons particulièrement à la méthode directe de support pour les réseaux, qui est une approche pour résoudre certains des problèmes les plus courants rencontrés dans les réseaux, tels que le problème du flot à coût minimum. Cette méthode utilise des techniques avancées de la théorie des graphes et de l'optimisation pour identifier les parties les plus importantes du réseau qui contribuent au flux de sortie.

En fin de compte, notre objectif est de fournir une analyse de la méthode directe de support pour les réseaux, en fournissant des exemples concrets pour illustrer son utilisation. Nous espérons que ce chapitre permettra de mieux comprendre l'importance de la théorie des réseaux et de fournir des outils pour résoudre des problèmes complexes dans ce domaine.

4.1 Méthode de support réseau

Considérons de nouveau le problème de flot à coût minimum sous la forme standard suivante :

$$\min_x Z = c^T x, \quad (4.1)$$

$$Ax = b, \quad (4.2)$$

$$0 \leq x \leq u, \quad (4.3)$$

ici tous les paramètres du problème (4.1) sont définis dans la section [3.1] et u_{ij} est la capacité des arcs.

Définition 4.1. Soit T un arbre. La base est constituée d'arcs $(i, j) \in T$ formant une arborescence recouvrante. La paire $\{x, T\}$ est dite solution réalisable de support (SRS) du problème (4.1) si x vérifie les contraintes (4.2)-(4.3) et aussi :

$$0 \leq x_{ij} \leq u_{ij}, \forall (i, j) \notin T, \quad \delta_{ij} = 0, \forall (i, j) \in T \quad (4.4)$$

Le but de l'algorithme est alors de construire une solution optimale x^0 . L'itération de l'algorithme consiste à passer de $\{x, T\}$ vers une autre SRS $\{\bar{x}, \bar{T}\}$ de telle sorte à assurer une amélioration de la valeur de la fonction objectif : $Z(\bar{x}) \leq Z(x)$.

A cet effet, on construit la nouvelle solution réalisable \bar{x} de la manière suivante :

$$\bar{x} = x + \theta l, \quad \theta \geq 0,$$

où l est un n -vecteur appelé direction d'amélioration, θ étant le pas toute au long de cette direction.

4.1.1 Vecteur des coûts réduits

$$\delta_{ij} = c_{ij} - y_i + y_j, \forall (i, j) \in E,$$

où c_{ij} représente le coût de l'arc (i, j) , y_i et y_j sont les variables duales associées aux nœuds i et j respectivement.

Si l'arc (i, j) est dans l'arborescence de la solution courante, alors δ_{ij} est défini comme 0, ce qui signifie que le coût réduit de cet arc est nul :

$$\delta_{ij} = 0, \forall (i, j) \in T.$$

4.1.2 Critère d'optimalité

Les relations suivantes décrivent les conditions suffisantes (et dans le cas de la non-dégénérescence, nécessaires) pour l'optimalité de la solution réalisable de support $\{x, T\}$:

$$\begin{cases} \delta_{ij} \geq 0, & \text{si } x_{ij} = 0, \\ \delta_{ij} \leq 0, & \text{si } x_{ij} = u_{ij}, \\ \delta_{ij} = 0, & \text{si } 0 \leq x_{ij} \leq u_{ij}. \end{cases}$$

Ces relations indiquent que si le flot sur l'arc (i, j) est à ses bornes inférieure (resp. supérieure), alors le coût réduit δ_{ij} doit respectivement être négatif ou (resp. positif). Si le flot sur l'arc (i, j) est à l'intérieur de ses bornes, le coût réduit δ_{ij} doit être nul.

En vérifiant ces conditions pour tous les arcs du réseau, on peut déterminer si la solution réalisable est optimale ou s'il est nécessaire d'apporter des ajustements au flot pour améliorer la solution.

Choix de la variable entrante

- $x_{ij} = 0$ et $\delta_{ij} < 0$, cela signifie que le coût réduit de l'arc (i, j) est négatif et que le flot actuel est en dessous de sa borne supérieure. Par conséquent, pour améliorer la solution, on augmente le flot sur l'arc (i, j) .
- $x_{ij} = u_{ij}$ et $\delta_{ij} > 0$, cela signifie que le coût réduit de l'arc (i, j) est positif et que le flot actuel est à sa borne supérieure. Dans ce cas, pour améliorer la solution, on diminue le flot sur l'arc (i, j) .
- $0 \leq x_{ij} \leq u_{ij}$ et $\delta_{ij} \neq 0$, cela signifie que le flot sur l'arc (i, j) est à l'intérieur de ses bornes et que le coût réduit est non nul. Dans ce cas, des ajustements doivent être effectués pour atteindre une solution optimale, mais la direction spécifique du changement dépendra de la structure du réseau.

Algorithm 3 Algorithme de support réseau

. Soit $\{x, T\}$ une SRS du problème (4.2)-(4.3) et x une solution réalisable vérifie . Le schéma de l'algorithme de la méthode de support réseau comporte les étapes suivantes :

(01) initialiser l'arbre couvrant T .

(02) introduire une solution réalisable SRS $x \in \mathbb{R}^{n \times m}$.

(04) Test d'optimalité de la SRS $\{x, T\}$

Si A) :

les critères d'optimalité suivant sont vérifiés :

$$\delta_{ij} \geq 0, \text{ si } x_{ij} = 0,$$

$$\delta_{ij} \leq 0, \text{ si } x_{ij} = u_{ij},$$

$$\delta_{ij} = 0, \text{ si } 0 \leq x_{ij} \leq u_{ij}.$$

Alors on arrête

Sinon B) :

(05) Amélioration de la solution réalisable x .

- Déterminer l'ensemble des indices non optimaux :

$$T_{Nopt} = \{(i, j) \notin T \mid (x_{ij} = 0 \text{ et } \delta_{ij} < 0) \vee (x_{ij} = u_{ij} \text{ et } \delta_{ij} > 0) \vee (0 < x_{ij} < u_{ij} \text{ et } \delta_{ij} \neq 0.)\}$$

- Choisir l'arc (i_0, j_0) tel que $|\delta_{i_0 j_0}| = \max\{|\delta_{ij}|, (i, j) \in T_{Nopt}\}$;

- Calculer le pas $\theta = \min\{\theta_{j_1}, \theta_{j_0}\}$, où θ_{j_1} et θ_{j_0} sont donnés par les relations (2.8) ;

- Calculer $\bar{x} = x + \theta l$, $Z(\bar{x})$;

(06) Changement de base \bar{x} :

- Si $\theta = \theta_{j_0}$, alors $\bar{T} = T$;

- Si $\theta = \theta_{j_1}$, alors $\bar{T} = (T \setminus j_1) \cup j_0$;

(07) Teste d'optimalité de la nouvelle SRS \bar{x} :

- Aller à (A) ;

- Sinon, aller à (B).

4.2 Exemple pratique

Considérons de nouveau l'exemple donné dans le chapitre précédent. Soit le réseau de transport suivant :

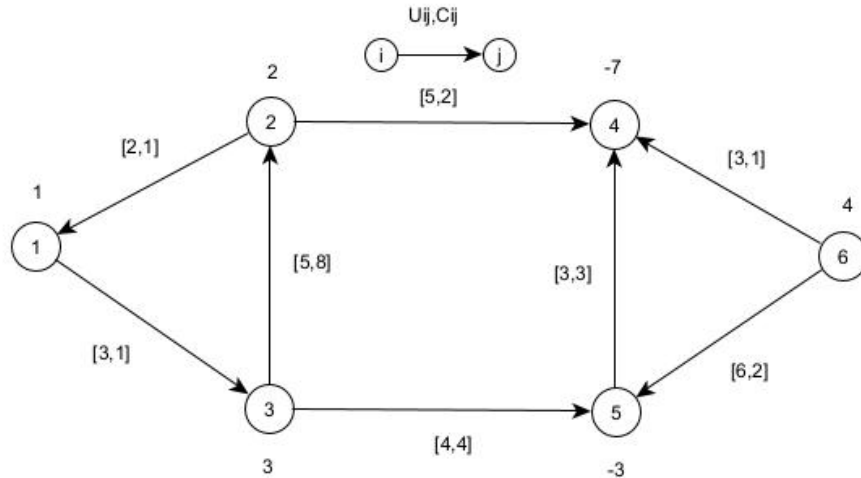


FIGURE 4.1 – Réseau de distribution

$$\text{Min} Z = x_{21} + x_{13} + 8x_{32} + 2x_{24} + 4x_{35} + 3x_{54} + x_{64} + 2x_{65}$$

$$S.c \left\{ \begin{array}{l} +x_{13} - x_{21} = 1 \\ +x_{21} - x_{32} + x_{24} = 2 \\ -x_{13} + x_{32} + x_{35} = 3 \\ -x_{24} - x_{54} - x_{64} = -7 \\ -x_{35} + x_{54} - x_{65} = -3 \\ x_{64} + x_{65} = 4 \end{array} \right.$$

$$0 \leq x_{21} \leq 2; 0 \leq x_{32}, x_{24} \leq 5; 0 \leq x_{35} \leq 4; 0 \leq x_{54}, x_{13}, x_{64} \leq 3; 0 \leq x_{65} \leq 6$$

Une solution réalisable : $x_{13} = 2; x_{21} = 1; x_{32} = 3; x_{24} = 4; x_{35} = 2; x_{54} = 2;$
 $x_{64} = 1; x_{65} = 3$

Considérons l'arbre recouvrant $T = (2, 1); (3, 2); (2, 4); (3, 5); (6, 5)$ la solution basique correspondante.

- **Iteration 1**

Calcul des vecteurs des coûts réduits basiques :

on pose $y_1 = 0$ avec $\delta = 0$

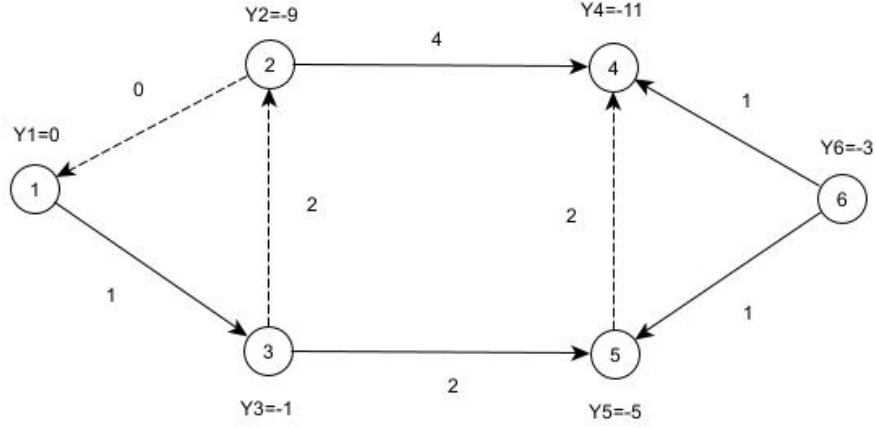


FIGURE 4.2 – Choix de l'arbre couvrant

$$\delta_{65} = c_{65} - y_6 + y_5 = 2 - y_6 + 5 = 0 \Rightarrow y_6 = 7$$

$$\delta_{35} = c_{35} - y_3 + y_5 = 4 - 9 + y_5 \Rightarrow y_5 = 5$$

$$\delta_{32} = c_{32} - y_3 + y_2 = 8 - y_3 + 1 \Rightarrow y_3 = 9$$

$$\delta_{24} = c_{24} - y_2 + y_4 = 2 - 1 + y_4 \Rightarrow y_4 = -1$$

$$\delta_{21} = c_{21} - y_2 + y_1 = 1 - y_2 + 0 \Rightarrow y_2 = 1$$

Calcul des vecteurs des coûts réduits non basiques :

$$\delta_{13} = c_{13} - y_1 + y_3 = 1 - 0 + 9 = 10 > 0 \text{ et } x_{13} = 2 \text{ Nopt}$$

$$\delta_{54} = c_{54} - y_5 + y_4 = 3 - 5 - 1 = -3 < 0 \text{ et } x_{54} = 2 \text{ Nopt}$$

$$\delta_{64} = c_{64} - y_6 + y_4 = 1 - 7 - 1 = -7 < 0 \text{ et } x_{64} = 1 \text{ Nopt}$$

- **Variable Entrante**

$$\delta_{ij} = \max_{(i,j) \in T_{Nopt}} \{|\delta_{ij}|\}, \text{ avec } T_{Nopt} = \{(i,j) \notin T : (x_{ij} = 0 \wedge \delta_{ij} < 0) \vee (x_{ij} = u_{ij} \wedge \delta_{ij} > 0)\}$$

$$\delta_{ij} = \max(10, |-7|, |-3|) = 10$$

Alors, on fait entrer l'arc (1, 3) en base pour former le cycle φ suivant :

$$- \theta^- = \min_{(i,j) \in \varphi^-} \{x_{ij}\} = \min\{x_{13}, x_{32}, x_{21}\} = \min\{2, 3, 1\} = 1$$

$$- \theta^+ = \min_{(i,j) \in \varphi^+} \{u_{ij} - x_{ij}\} = \infty, \text{ car } \varphi^+ = \emptyset$$

$$- \theta = \min\{\theta^+, \theta^-\} = 1$$

- **Variable sortante**

L'arc (2, 1) sort de la base, donc la base devient $T = T \cup (1, 3) \setminus (2, 1)$.

$$T = \{(1, 3); (3, 2); (2, 4); (3, 5); (6, 5)\}$$

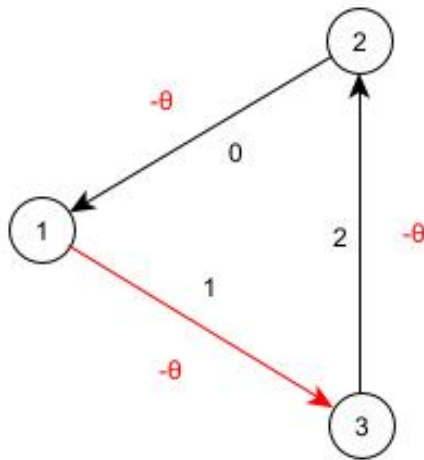


FIGURE 4.3 – Variable entrante

• **Iteration 2**

On vas vérifier les conditions d’optimalité pour le nouveau marquage.

Calcul des vecteurs des coûts réduits basiques :

- $\delta_{13} = c_{13} - y_1 + y_3 = 1 - 0 + y_3 \Rightarrow y_3 = -1$
- $\delta_{32} = c_{32} - y_3 + y_2 = 8 + 1 + y_2 \Rightarrow y_2 = -9$
- $\delta_{35} = c_{35} - y_3 + y_5 = 4 + 1 + y_5 \Rightarrow y_5 = -5$
- $\delta_{24} = c_{24} - y_2 + y_4 = 2 + 9 + y_4 = 0 \Rightarrow y_4 = -11$
- $\delta_{65} = c_{65} - y_6 + y_5 = 2 - y_6 - 5 = 0 \Rightarrow y_6 = -3$

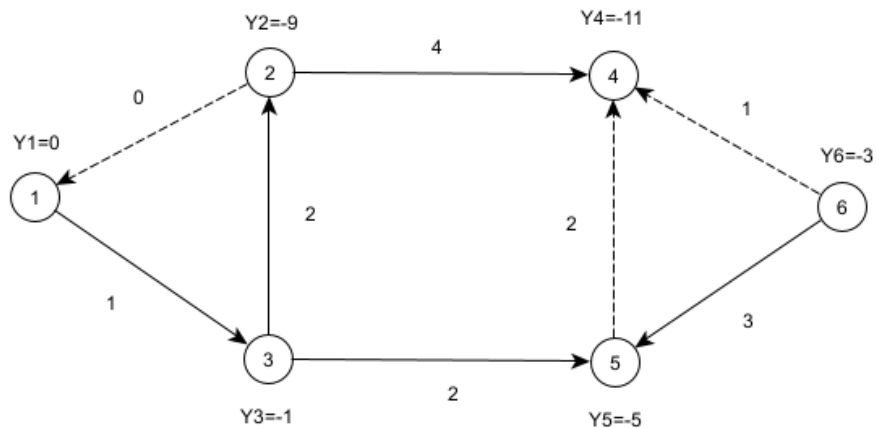


FIGURE 4.4 – Le nouveau marquage

Calcul des coûts réduits non basiques :

- $\delta_{21} = c_{21} - y_2 + y_1 = 1 + 9 + 0 = 10$ et $x_{21} = 0$ opt
- $\delta_{54} = c_{54} - y_5 + y_4 = 3 + 5 - 11 = -3 < 0$ et $x_{54} = 2$ Nopt
- $\delta_{64} = C_{64} - y_6 + y_4 = 1 + 3 - 11 = -7 < 0$ et $x_{64} = 1$ Nopt

• **Variable Entrante**

$$\delta_{ij} = \max\{|\delta_{ij}|, (i, j) \in T_{Nopt}\}, \text{ avec } T_{Nopt} = \{(i, j) \notin T : (x_{ij} = 0 \wedge \delta_{ij} < 0) \vee (x_{ij} = u_{ij} \wedge \delta_{ij} > 0)\}$$

$$\delta_{ij} = \max(|-7|, |-3|) = 7$$

L'arc (6,4) entre à la base T.

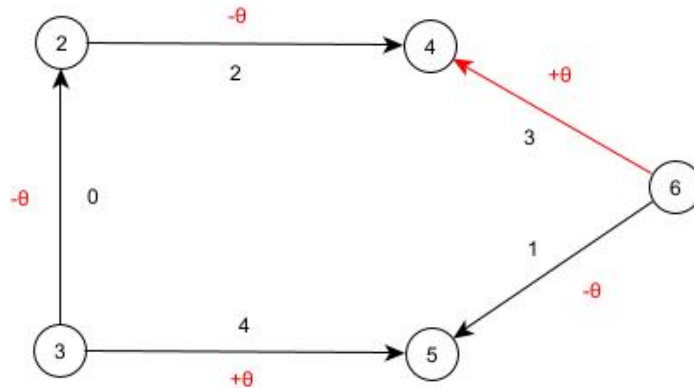


FIGURE 4.5 – Variable entrante

- $\theta^- = \min_{(i,j) \in \varphi^-} \{x_{ij}\} = \min\{x_{24}, x_{32}, x_{65}\} = \min\{4, 2, 3\} = 2$
- $\theta^+ = \min_{(i,j) \in \varphi^+} \{u_{ij} - x_{ij}\} = \min\{u_{64} - x_{64}, u_{35} - x_{35}\} = \min\{3 - 1; 4 - 2\} = 2$
- $\theta = \min\{2, 2\} = 2$

• **Variable sortante**

L'arc (3,2) sort de la base, donc la base T change.

$$T = \{T \cup (6, 4) \setminus (3, 2)\}$$

• **Itération 3**

Calcul des vecteurs des coûts réduits basiques :

— $\delta_{13} = c_{13} - y_1 + y_3 = 1 - 0 + y_3 \Rightarrow y_3 = -1$

— $\delta_{24} = c_{24} - y_2 + y_4 = 2 - y_2 - 4 \Rightarrow y_2 = -2$

— $\delta_{35} = c_{35} - y_3 + y_5 = 4 + 1 + y_5 \Rightarrow y_5 = -5$

— $\delta_{64} = c_{64} - y_6 + y_4 = 1 + 3 + y_4 \Rightarrow y_4 = -4$

— $\delta_{65} = c_{65} - y_6 + y_5 = 2 - y_6 - 5 \Rightarrow y_6 = -3$

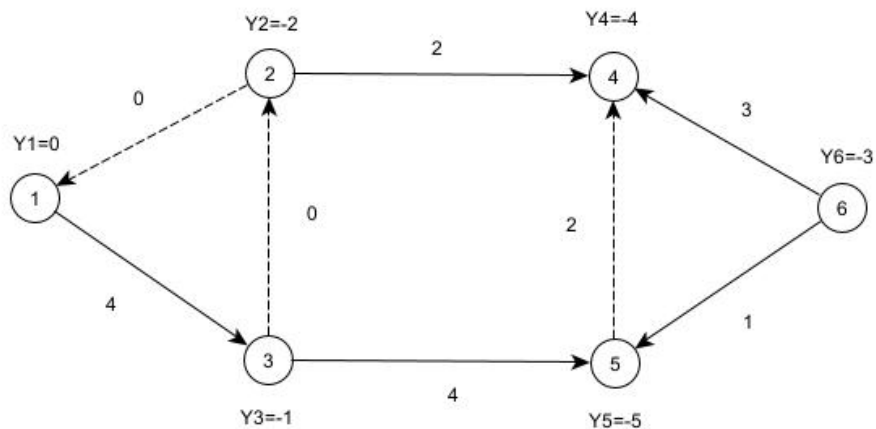


FIGURE 4.6 – Mise à jour des potentiels

Calcul des vecteurs des coûts réduits non basiques :

— $\delta_{21} = c_{21} - y_2 + y_1 = 1 + 2 + 0 = 3$ et $x_{21} = 0$ opt

— $\delta_{54} = c_{54} - y_5 + y_4 = 3 + 5 - 4 = 4 > 0$ et $x_{54} = 2$ Nopt

— $\delta_{32} = c_{32} - y_3 + y_2 = 8 + 1 - 2 = 7 > 0$ et $x_{64} = 0$ opt

• **Variable Entrante**

Le seul arc qui ne vérifie pas le critère d'optimalité est l'arc (5,4) donc il entre à la base T.

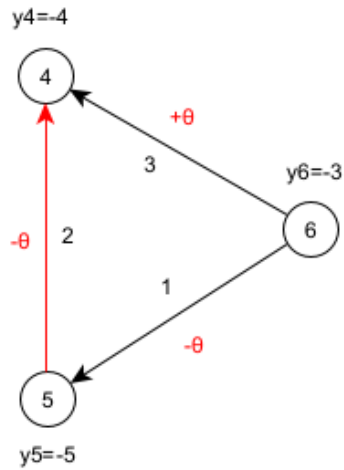


FIGURE 4.7 – La variable entrante

- $\theta^- = \min_{(i,j) \in \varphi^-} \{x_{ij}\} = \min\{x_{54}, x_{65}\} = \min\{2, 1\} = 1$
- $\theta^+ = \min_{(i,j) \in \varphi^+} \{u_{ij} - x_{ij}\} = \min\{u_{64} - x_{64}\} = \min\{3 - 3\} = 0$
- $\theta = \min\{1, 0\} = 0$

• **Itération 4**

Calcul des vecteurs des coûts réduits basiques :

- $\delta_{13} = c_{13} - y_1 + y_3 = 1 - 0 + y_3 \Rightarrow y_3 = -1$
- $\delta_{24} = c_{24} - y_2 + y_4 = 2 - y_2 - 8 \Rightarrow y_2 = -6$
- $\delta_{35} = c_{35} - y_3 + y_5 = 4 + 1 + y_5 \Rightarrow y_5 = -5$
- $\delta_{54} = c_{54} - y_5 + y_4 = 1 + 3 + y_4 \Rightarrow y_4 = -8$
- $\delta_{64} = c_{64} - y_6 + y_4 = 1 - y_6 - 8 \Rightarrow y_6 = -3$

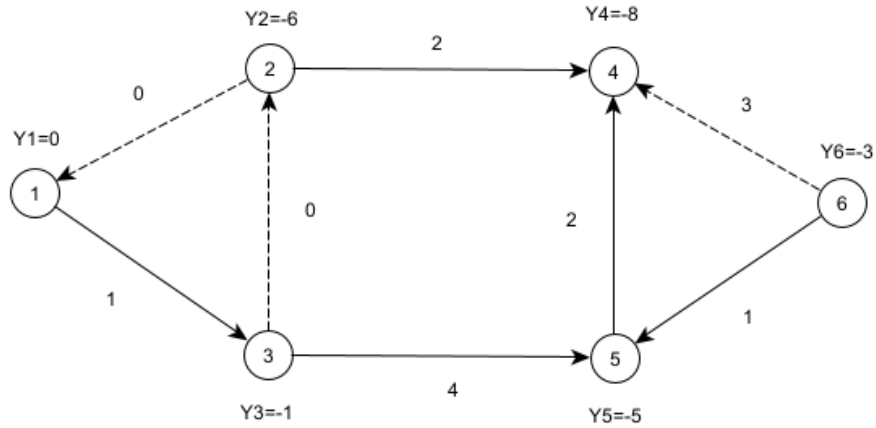


FIGURE 4.8 – La solution optimal

Calcul des vecteurs des coûts réduits non basiques :

- $\delta_{21} = 7$ et $x_{21} = 0$ opt
- $\delta_{32} = 3$ et $x_{32} = 0$ opt
- $\delta_{64} = -4$ et $x_{64} = 3 = u_{64}$ opt

Par conséquent, la solution optimale est :

$$x^* = (x_{21} = 0; x_{13} = 1; x_{32} = 0; x_{24} = 2; x_{35} = 4; x_{54} = 2; x_{64} = 3; x_{65} = 1)$$

Le coût total minimal :

$$Z^*(x) = 1 \times 0 + 1 \times 1 + 8 \times 0 + 2 \times 2 + 4 \times 4 + 3 \times 2 + 1 \times 3 + 2 \times 1$$

$$Z^*(x) = 32$$

4.3 Exemple d'application

Le problème de distribution que nous considérons ici implique un usine, deux plate-formes de stockage et six clients à approvisionner, tel que schématisé sur la figure suivante :

L'objectif est de déterminer comment acheminer le produit de l'usine vers les clients de la manière la plus efficace possible, et ce en minimisant les coûts de transport et tout en respectant les contraintes de capacité de transport sur les différentes liaisons.

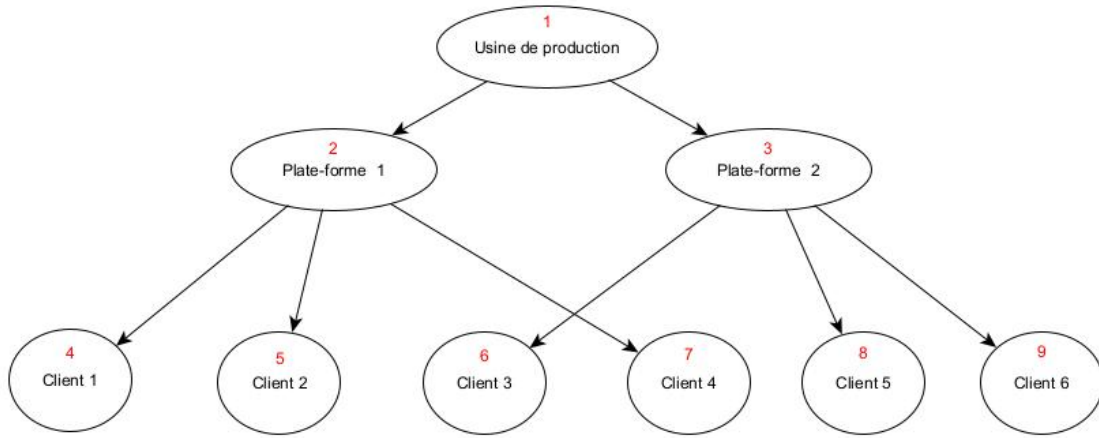


FIGURE 4.9 – Réseau de distribution de l'entreprise

4.3.1 Construction du modèle

Le problème de distribution de la figure(4.1) est représenté par le réseau de ransport $G = (V, E, u, c)$ où :

- **V** : Les noeuds du réseau représentant, l'usine de production, les deux plate-formes et les six clients ; avec $n = |V| = 9$, La signification des indices des sommets est donnée comme suit :
 - Pour $i = 1$: L'usine de production ;
 - pour $i = 2, 3$: Les plate-formes 1 et 2 respectivement ;
 - Pour $i = 4 \dots 9$: Les clients 1 jusqu'à 6.

- **Paramètres**

b_i : L'offre et la demande au niveau des noeuds ;

u_{ij} : Les capacités de transport sur les arcs ;

c_{ij} : Coût de transport du produit sur l'arc (i, j) ; u_{12} : La capacité de transport de l'usine de production vers la plate-forme 1.

- **Variable de décision :**

x_{ij} : La quantité à acheminer sur l'arc (i, j) ;

Dans notre étude on s'est focalisés pour le système de distribution sur un seul produit. Nous avons modélisé ce problème de distribution comme étant un problème généralisé du flot maximum à coûts minimum.

- **Fonction objectif**

L'objectif principal pour ce problème est de minimiser le coût de transport entre le centre de

production et les plates-formes, ainsi que le coût de transport entre les plates-formes et les clients. En d'autres termes, il s'agit de minimiser le coût total de transport.

• **Contraintes**

1. Les quantités acheminés à partir de l'usine vers les plates-formes ne doivent pas dépasser la capacité de production du produit au niveau de l'usine.

$$x_{12} + x_{13} = b_1.$$

2. Les quantités acheminées à partir de la plate-forme 2 vers les clients 4, 5, 7 :

$$x_{24} + x_{25} + x_{27} = b_2.$$

3. Les quantités acheminés à partir de plate-forme 3 vers les clients 6, 8, 9 :

$$x_{36} + x_{38} + x_{39} = b_3.$$

Donc, le PFCM associé au problème formulé par PL suivant :

$$\begin{cases} \min Z = \sum_{(i,j) \in E} c_{ij} x_{ij} \\ Ax = b, \\ 0 \leq x_{ij} \leq u_{ij}, \forall (i, j) \in E. \end{cases} \quad (4.5)$$

Sous forme matricielle, le modèle devient :

$$\begin{cases} \min Z = c^T x \\ Ax = b \\ 0 \leq x \leq u \end{cases} \quad (4.6)$$

Les données associées au modèle (4.6)

| | | | | | | | | |
|-------------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Arc (i, j) | (1, 2) | (1, 3) | (2, 4) | (2, 5) | (2, 7) | (3, 6) | (3, 8) | (3, 9) |
| Coût c_{ij} | 60 | 70 | 50 | 40 | 50 | 50 | 35 | 40 |
| Capacité u_{ij} | 120 | 100 | 55 | 45 | 60 | 50 | 40 | 30 |

TABLE 4.1 – Coûts et capacités des arcs

| | | | | | | | | | |
|---------------|------|---|---|----|----|----|----|----|----|
| Noeud i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Demande b_i | -200 | 0 | 0 | 40 | 40 | 30 | 30 | 30 | 30 |

TABLE 4.2 – Demandes des noeuds

• Résultat

Les résultats obtenue après l'exécution de programme :

| Arc (i, j) | (1, 2) | (1, 3) | (2, 4) | (2, 5) | (2, 7) | (3, 6) | (3, 8) | (3, 9) |
|---------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Flot x_{ij} | 110 | 90 | 40 | 40 | 30 | 30 | 30 | 30 |

TABLE 4.3 – Le flot associe à chaque arcs

4.3.2 Interprétation des résultats

L'objet de notre modèle présenter dans la section 4.4.1 est de minimiser les coûts de transports de l'usine vers les plate-formes ainsi que de plate-formes vers les clients ce que montre l'exemple didactique après son implantation et exécution sur le logiciel MATLAB.

Voici la signification des différentes matrices et résultats :

- Matrice u : Cette matrice représente les capacités maximales des arcs dans le réseau. Par exemple, $u_{12} = 120$ indique que l'arc entre le nœud 1 et le nœud 2 a une capacité maximale de 120.
- Vecteur b : Ce vecteur représente les demandes des clients pour les produits. Chaque valeur correspond à la quantité demandée par un client spécifique. Par exemple, $b_1 = -200$ indique que le client 1 demande une quantité de -200 (ce qui peut signifier une quantité négative de produits à retourner).
- La fonction "simplex(u, b, c)" effectue l'algorithme du simplexe réseau pour résoudre le problème de flux maximal dans le réseau en tenant compte des capacités, des coûts et des demandes spécifiés. Le résultat de cette fonction, "minf", est un objet qui représente la solution optimale du problème.
- Matrice c : Cette matrice représente les coûts de transport entre les nœuds du réseau. Chaque valeur correspond au coût associé à l'acheminement d'un produit d'un nœud source à un nœud de destination. Par exemple, $g_{23} = 70$ indique que le coût de transport du nœud 2 au nœud 3 est de 70.
- Matrice minf : Cette matrice représente la solution optimale obtenue par l'algorithme de résolution. Elle indique les quantités de produits à transférer le long de chaque arc du réseau

pour minimiser les coûts totaux de distribution. Les valeurs non nulles dans la matrice minf représentent les quantités de produits à transférer.

- Valeur totale : La valeur totale, indiquée par $ans = 21750$, représente le coût total associé à la solution optimale. C'est la somme des coûts de transport pour toutes les quantités de produits transférées.

Conclusion

Dans ce chapitre, nous avons étendu la méthode directe de support de la programmation linéaire classique au problème de flot à coût minimum. Le schéma de l'algorithme de support réseau est donné, puis nous l'avons appliqué à un exemple concret avec une implémentation sur MATLAB .

CONCLUSION GÉNÉRALE

Notre étude s'est focalisée sur le problème de flot à coût minimum dans les réseaux. L'objectif principal de ce travail est d'appliquer la méthode de support pour résoudre ce problème.

Pour ce faire, nous avons rappelé dans un premier lieu les bases théoriques nécessaires pour aborder ce sujet. Cela inclut des notions sur la théorie des graphes, une introduction à l'optimisation dans les réseaux, ainsi que la présentation de la méthode directe de support de résolution de PLs sous forme standard. Par la suite, nous avons présenté l'algorithme du simplexe réseau [9] qui est spécifiquement dédiée à la résolution du PFCM. Cet algorithme est très efficace et permet dans nos jours de résoudre des problèmes pratiques ayant des milliers de noeuds et d'arcs (problèmes de grande taille).

La méthode directe de support [13] est l'une des méthodes développées pour résoudre des problèmes de programmation linéaire. Cette approche a la particularité de traiter les problèmes pratiques tels qu'ils se présentent et utilise le critère de sub-optimalité pour arrêter le processus de résolution à une précision donnée par le décideur. Elle a aussi l'avantage de démarrer à partir de n'importe quelle solution réalisable qui peut être un point extrême ou un point intérieure, contrairement à la méthode du simplexe qui utilise que des points extrêmes.

Dans ce travail, nous avons étendu la méthode de support classique au PFCM et on a schématisé les étapes de l'algorithme de support réseau. Par la suite, nous avons modifié le code de l'algorithme du simplexe réseau [18] pour implémenter notre approche sur Matlab. L'étude se termine par un exemple pratique sur la logistique, qu'on a modélisé comme étant un PFCM et qu'on a résolu en utilisant l'algorithme de support réseau.

En conclusion, la méthode de support réseaux se révèle être une approche prometteuse pour résoudre efficacement des problèmes d'optimisation dans les réseaux et plus particulièrement le

problème de flot à coût minimum. Cependant, il reste encore des possibilités d'amélioration et d'extension de ce travail, à savoir :

- Application de la méthode adaptée pour la résolution du PFCM, puis traiter la version multiobjectif du problème.

Résumé

Le problème de flot à coût minimum est l'un des problèmes classiques d'optimisation dans les réseaux. Ce dernier modélise diverses situations pratiques et qui sont résolues efficacement par la méthode du simplexe réseau. Le but de notre travail est d'appliquer la méthode directe de support à ce problème, et ce en tenant compte de la spécificité du problème traité. L'algorithme du support réseau proposé est itératif et permet de résoudre des problèmes d'optimisation dans les réseaux. Le schéma de l'algorithme est donné, puis illustré par un exemple pratique en logistique. Ce cas concret est modélisé comme étant un problème de flot à coût minimum, puis il est résolu par l'algorithme de support que nous avons implémenté sur Matlab. **Mots clés :** Problème de flot à coût minimum, Optimisation dans les réseaux, Méthode simplexe réseaux, Méthode de support réseau.

Abstrat

The minimum-cost flow is one of the classic network optimization problems. It models a variety of practical situations that are efficiently solved by the network simplex method. The aim of our work is to apply the direct support method to this problem, taking into account the specificity of the problem at hand. The proposed network support algorithm is iterative and can be used to solve optimization problems in networks. A description of the algorithm is given, followed by a practical example in logistics. This concrete case is modeled as a minimum-cost flow problem, then solved by the support algorithm that we have implemented on Matlab.

Keywords : Minimum cost flow problem, Network optimization, Network simplex method, Network support method.

- [1] Andre R. *De l'optimisation dans les réseaux*. Mémoire HDR, Université de Bretagne-Sud, Ecole doctorale SICMA, France, 2012 .
- [2] Brahmi B. *Optimisation dans les réseaux*. Cours Master 2 MMEPR, Université de Béjaia, 2023.
- [3] Bertsekas. Dimitri P. *A Unified Framework for Primal-dual Methods in Minimum Cost Network Flow Problems*. Mathematical Programming, vol 32, 125-145, 1985.
- [4] Bibi. M. O. *Programmation linéaire et quadratique*. Cours Master 1 MMEPR, Université de Béjaia, 2021.
- [5] Boudjema W. et Bouras S. *Optimisation dans les réseaux* . Mémoire de fin de cycle, université de Bejaia, 2016.
- [6] Cohen J .Théorie des graphes et algorithmes. (oct 2006).
- [7] Didier M. *Introduction à la théorie des graphes*, Cahier de La CRM, Numéro 6, 2012.
- [8] Desrosiers J., Pelleitier P., et Soumis F., *Plus court chemin avec contraintes d'horaire*. RAIRO, Recherche opérationnelle, vol 17 (4), 1983.
- [9] Dantzig G.B., *Linear programming and extensions*. Princeton University Press, Princeton, New Jersey, 1963
- [10] Edmond. *Algorithmes pour le problème d'arbre couvrant minimale*. Mémoire de maîtrise en informatique, université de Québec, 2014.
- [11] Julie S. *Résolution de problèmes d'optimisation pour les réseaux de transport d'électricité de grande taille avec des méthodes de programmation semi-définie positive*. Thèse de doctorat en cotutelle, Université Sorbonne Paris Nord, 2021.
- [12] Gandron M., et Minoux M. *"Graphes et algorithmes"*. Editions Eyrolles, Paris VI 1986.

- [13] Gabasov R., Kirillova F.M., Kostukova O.I, *Théorie constructive des problèmes extrémaux*. Minsk Universitetskoyé, 1984.
- [14] Kiraly Z.et Kovacs P, Efficient implementations of minimum-cost flow algorithms, Acta Univ.Sapientiae, Informatica ,2012
- [15] Mebareki S et Tahir L. *Optimisation du réseau logistique de distribution :cas des huiles au niveau de Cevital*. Mémoire de fin de cycle, université de Bejaia, 2014-2015.
- [16] Mounsi T. *Théorie des graphes et optimisation*. Notes de cours, Institut Supérieur d'Informatique et de Multimédia Sfax, Septembre 2014.
- [17] Matlab. <http://fr.mathworks.com/products/matlab>. Consulter le :21-05- 2019.
- [18] Naomichi A. *The Network Simplex Algorithm*. <https://www.mathworks.com/matlabcentral/fileexchange/29665-the-network-simplex-algorithm>, 2023.
- [19] Roseaux R. *Exercices et problèmes résolus de recherche opérationnelle : programmation linéaire et extensions*. Édition 3, DUNOD, Paris, 1985.
- [20] Ravindra K., Magnanti L., and Orlin B. *Networks flows*. Massachusetts institute of technology, cambridge, 1988.
- [20] Ravindra K., Magnanti., Ahuja., Thomas L., et James B. in Handbooks *Handbooks in Operations Research and Management Science, 1 Optimization*. North-Holland,(1989).
- [21] Susann S. *Contributions à la conception de réseau de service en transport*. Thèse de doctorat, Ecole Doctorale I-MEP2, université de Grenoble, 2010.
- [22] Sébastien L. et Alain H. *Fondements de recherche opérationnelle*. Notes de cours, École Polytechnique de Montréal, 2020.
- [23] Solnon C. *Théorie des graphes et optimisation dans les graphes*. Notes de cours, Université de Lyon,2016.
- [24] Taouinet S., *Théorie des Graphes avancés*. Note de cours, Université de Béjaia, 2016.
- [25] Zoltàn S. *Recherche opérationnel, cours master 1 mathématique et applications* . Université de Grenoble, France, 2020.